jamk.fi

IEEE 802.1X JA PORTTIKOHTAINEN TODENNUS WINDOWS-YMPÄRISTÖSSÄ

Anssi Kinnunen

Opinnäytetyö
Toukokuu 2014

Tietotekniikan koulutusohjelma
Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

KUVAILULEHTI

Tekijä(t) Kinnunen, Anssi	Julkaisun laji Opinnäytetyö	Päivämäärä 14.5.2014
	Sivumäärä 86	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty (X)

Työn nimi
IEEE 802.1X JA PORTTIKOHTAINEN TODENNUS WINDOWS-YMPÄRISTÖSSÄ

Koulutusohjelma
Tietotekniikan koulutusohjelma

Työn ohjaaja(t)
Saharinen, Karo
Piispanen, Juha

Toimeksiantaja(t)
Betset Oy
Lillvis, Jarno

Tiivistelmä

Opinnäytetyö tehtiin Betset Oy:n tietohallinnolle. Työn tarkoituksena oli perehtyä IEEE 802.1X-standardin toimintaan, mahdollisuuksiin sekä toteuttaa sen avulla tilaajan verkkoympäristöön porttikohtainen todennus. Työn teoriaosuudessa tutkittiin kyseisen järjestelmän käyttämiä verkkotekniikoita kuten EAP-metodeita, Ethernetiä ja RADIUS-protokollaa. Lisäksi käsiteltiin pintapuolisesti Windows-työympäristön rakennetta ja työssä vaadittuja palveluita.

Työn käytännön osuudessa rakennettiin porttikohtaisen todennuksen vaatima ympäristö. Tämä tapahtui asentamalla kahdennetut Microsoft NPS-palvelimet, konfiguroimalla verkon aktiivilaitteet ja asettamalla työasemat todennuksen alle. Käytetty todennusmenetelmä oli varmenteisiin perustuva EAP-TLS ja sen toimintaa käsiteltiin tarkemmin työn teoriaosuudessa.

Testaamisvaiheessa pyrittiin selvittämään järjestelmän luotettavuus ja toiminta tilaajan tuotantoympäristössä. Ongelmia tuli työasemien osalta, kun testattiin toissijaista NPS-palvelinta ja sen vaikutuksia työympäristöön. Kyseiset ongelmat saatiin ratkaistua tekemällä tarvittavat muutokset työasemien asetuksiin. Lopuksi tarvittavat asetukset otettiin käyttöön kaikissa työasemissa käyttämällä Windowsin ryhmäkäytäntöobjektia.

Työ onnistui vaatimusten mukaisesti, ja lopputuloksena tilaaja sai luotettavan sekä helposti hallittavan järjestelmän. Tämä työ helpotti tietohallinnon työmäärää, sillä myös muiden paikkakuntien toimipisteet saatiin saman järjestelmän hallintaan. Lisäksi työ antoi kattavan dokumentaation kyseisen järjestelmän toiminnasta sekä käytettävistä tekniikoista.

Avainsanat (asiasanat)

IEEE 802.1X, EAP-TLS, NPS, RADIUS

Muut tiedot

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

DESCRIPTION

Author(s)	Type of publication	Date
Kinnunen, Anssi	Bachelor´s Thesis	14.4.2014
	Pages 86	Language Finnish
		Permission for web publication (X)

Title
IEEE 802.1X AND PORT-BASED AUTHENTICATION IN WINDOWS ENVIROMENT

Degree Programme
Information Technology

Tutor(s)
Saharinen, Karo
Piispanen, Juha

Assigned by
Betset Oy
Lillvis, Jarno

Abstract

The purpose of this bachelor's thesis was to become familiar with the IEEE 802.1X standard, its activities, and opportunities and to create with it port-based authentication for the company's network environment. Along with the necessary network protocols and network technologies this bachelor's thesis gives a light overview of the structure and necessary services of Windows Server operating system.

After the introductory briefing, the thesis continues with the practical implementation and testing of the new system. The purpose of the thesis was to find and solve all conflicts between the new and current work environment that have come up with the port-based authentication.

Finally, the thesis creates a detailed documentation of this research project for the assigning company. Based on the documentation, bachelor's thesis presents the end result of the project.

Keywords

IEEE 802.1X, EAP-TLS, NPS, RADIUS

Miscellaneous

Sisältö

Kuviot

Taulukot

Lyhenteet ja termit

ATM	Asynchronous Transfer Mode. Vanha pakettikytkentäinen tiedonsiirtoprotokolla.
AD	Active Directory. Windows-toimialueen käyttäjätietokanta ja hakemistopalvelu, joka sisältää tietoa käyttäjistä, tietokoneista ja tietoverkon resursseista.
ARP	Address Resolution Protocol. Protokolla, jonka tehtävä on selvittää IP-osoitteita.
CA	Certificate Authority. Varmentaja on luotettu taho, joka myöntää varmenteita käyttäjien, työasemien ja palveluiden käyttöön.
Child domain	Alitoimialueesta käytetty nimitys.
Collision	Kilpavarausmenettelyssä syntyvä tilanne, jossa kahden tai useamman työaseman lähetykset menevät päällekkäin.
DC	Domain Controller. Windows-toimialueen ohjainkone.
Detection	Lähiverkossa tapahtuneen törmäyksen havainnointi.
Domain	Toimialue on joukko Windows-työasemia, joita hallinnoidaan keskitetysti ja joista vähintään yksi on palvelin.
DNS	Domain Name System. Nimipalvelujärjestelmä, joka muuntaa mm. verkkotunnuksia IP-osoitteiksi.
EAP	Extensible Authentication Protocol. Yksinkertainen todentamisprotokolla, kehitetty alun perin PPP-protokollan kanssa.
EAPOL	Extensible Authentication Protocol Over LAN. Paketointitekniikka, jonka avulla siirretään EAP-kehyksiä.
Ethernet	Pakettipohjainen lähiverkkotekniikka.

Forest	Metsä. Windows-toimialueen ylin taso.
Frame	Kehys. Ethernet-verkossa siirrettävä datapaketti, mihin kuljetettava tieto pakataan.
Frame Relay	Vanhentuva lähiverkkojen yhdistämisessä käytetty protokolla.
FCS	Frame Check Sequence. Varmistussumma on tiedonsiirrossa käytetty menetelmä tiedon eheyden takaamiseksi.
GP	Group Policy. Ryhmäkäytäntö on tehokas työkalu työasemien ja käyttäjien hallitsemiseen.
GPMC	Group Policy Management Console mahdollistaa ryhmäkäytänteiden hallinnan.
GPO	Group Policy Object. Kokoelma tallennettuja ryhmäkäytäntöjä.
IEEE 802.1X	IEEE:n standardi, joka määrittää porttikohtaisen todennuksen.
IEEE	The Institute of Electrical and Electronics Engineers. Kansainvälinen tekniikan alan järjestö.
IP	Internet Protocol. Käytetään tietoliikennepakettien kuljettamisessa pakettikytkentäisessä tietoliikenneverkossa.
ISO	International Organization for Standardization. Kansainvälinen standardisoimisjärjestö.
iSCSI	Internet Small Computer System Interface. IP-verkoissa käytettävä protokolla, jolla voidaan liittää tallennuslaitteita ja järjestelmiä toisiinsa.
IETFS	The Internet Engineering Task Force. Internet-protokollien standardoinnista vastaava organisaatio, jonka tarkoituksena on tuottaa Request for comment -standardeja.
Kytkin	Lähiverkon aktiivilaite, joka toimii yleensä OSI-mallin toisella kerroksella välittäen datapaketteja MAC-osoitteiden peruseteella.

LAN	Local Area Network eli lähiverkko. Paikallisen alueen tietoverkosta käytetty nimitys.
LLC	Logical Link Control. Lähiverkkostandardissa määritetty siirtoyhteyskerros.
MAC	Media Access Control. Verkkorajapintojen yksilöintiin suunniteltu osoitteistus ja väylänvarauksesta käytetty termi.
MMC	Microsoft Management Console. Windowsin hallintatyökalu, jonka lisäosilla voidaan vaikuttaa mm. käyttöjärjestelmän toimintaan.
NFS	Network File System. Tiedostonsiirtoprotokolla, jonka avulla voidaan siirtää tiedostoja järjestelmästä toiseen.
NAS	Network Access Server. Tietoverkon liityntäpiste.
NPS	Network Policy Server. Windowsin versio RADIUS-palvelimesta.
OU	Organizational units. Active Directoryn niin sanottu säiliö, joka pitää sisällään käyttäjä- tai tietokoneobjekteja.
OSI	Open Systems Interconnection Reference Model. Kuvaa tiedonsiirtoprotokollien yhteistoiminnot seitsemän kerroksen mallina.
Palvelin	Työasema, jonka tehtävänä on tarjota erilaisia palveluja muille ohjelmille, joko tietoverkon välityksellä tai paikallisesti.
Payload	Hyötykuorma. Nimitys tiedonsiirtoprotokollien kentälle, joka on tarkoitettu tiedon kuljettamiseen.
Preamble	Alkutahdistus. Nimitys tiedonsiirtoprotokollan kentälle, joka on tarkoitettu osoittamaan muun muassa Ethernet-kehyksen alkamiskohta.
Padding	Täyte. Nimitys tiedonsiirtoprotokollan kentälle, joka on tarkoitettu täydentämään vajaamittaiset kehykset.

PKI Public Key Infrastructure. Julkisen avaimen infrastruktuuri määrittä palvelimet ja palvelut joiden avulla hallitaan varmenteita sekä julkisia avaimia.

PPP Point-to-Point Protocol. Protokolla, jota käytetän muodostaessa suora yhteys kahden osapuolen välille.

Paketti Tiedonsiirrossa käytettävä datayksikkö, joka sisältää hyötydatan ohella mm. lähde- ja kohdeosoitteet.

Peilaus Verkkotekniikassa käytetty termi, joka tarkoittaa verkkoliikenteen monistamista ja kopioimista toiseen verkkorajapintaan.

RADIUS Remote Authentication Dial In User Service. Yleisin todennusprotokolla.

RFC Request For Comments. IETF-organisaation julkaisema standardi.

Root domain Toimialueen ylin nimiavaruus.

RSoP Resultant Set of Policy. Microsoftin työkalu ryhmäkäytäntöobjektien tarkasteluun.

Tree Puu. Toimialueiden hierarkkisesta rakenteesta käytetty nimitys.

VLAN Virtual Local Area Network. IP-tasolla olevan verkkoliikenteen leimaustekniikka, joka mahdollistaa useamman loogisen lähiverkon muodostamisen.

1 Työn lähtökohdat

1.1 Toimeksianto ja tavoitteet

Opinnäytetyön tarkoituksena oli perehtyä standardin IEEE 802.1X toimintaan, mahdollisuuksiin sekä toteuttaa sen perusteella tilaajan verkkoympäristöön porttikohtainen todennus.

Tavoitteena oli saada aikaiseksi toimiva verkkoympäristö, jossa kaikki työasemat joutuivat todistamaan jäsenyytenä yrityksen toimialueeseen. Lisäksi kyseisen menetelmän tuli olla mahdollisimman käyttäjäystävällinen sekä varmatoiminen kaikissa tilanteissa.

1.2 Betset Oy

Betset Oy on yksi Suomen suurimpia betonisia valmisosia tuottava betonialan perheyritys ja se on perustettu Kyyjärvellä vuonna 1950. Ensimmäiset tuotteet olivat yksinkertaisia kaivo- ja rumpurenkaita sekä kattotiiliä. Tästä lähtien yritys on laajentunut tasaisesti ja samalla lisännyt tuotevalikoimaansa rakennusalan tarpeiden mukaan. (Historia 2014.)

1970-luvulla rakentaminen vilkastui huomattavasti, Betset vastasi tähän haasteeseen kehittämällä tuotantoaan valmisbetonille sopivaksi ja vuosikymmenen lopussa tuotevalikoimaan tulivat ensimmäiset betonielementit. Sukupolvenvaihdoksen myötä kehitys ja tuotevalikoima kasvoivat hurjasti ennen vuosituhannen vaihdosta. Tuotevalikoimaan tulivat mm. ontelo- ja kuorilaatat sekä jännebetonielementit. Uuden tekniikan ansiosta Betset kykenee valmistamaan Suomen järeimmät jännebetonielementit. (Historia 2014.)

Vuosituhannen vaihteen jälkeen toiminta laajeni Hämeenlinnaan, Nurmijärvelle ja Helsinkiin vahvistaen yrityksen markkina-asemia Etelä-Suomessa. Näillä toimipisteillä valmistetaan mm. valmisbetonia sekä suurten rakennusprojektien vaatimia seinä-, ontelo- ja TT-elementtejä. (Historia 2014.)

Viimeisin laajennus oli vuonna 2009 Venäjän Pietariin perustettu ZAO Betset, jossa valmistetaan kantavia väliseiniä Venäjän kasvaville markkinoille. Tuotanto Betset Oy:n nimissä aloitettiin tammikuussa 2010. (Historia 2014.)

Betset on merkittävä työllistäjä kotipaikkakunnallaan Kyyjärvellä, jossa työntekijöitä on noin 170. Hämeenlinnan tehdas työllistää noin 20 työntekijää, Nurmijärven tehdas puolestaan 60, Helsingin valmisbetonitehdas 5 ja Pietarin tehdas noin 140 työntekijää. (Heikkilä 2013.)

2 Lähiverkkotekniikka

2.1 OSI-malli

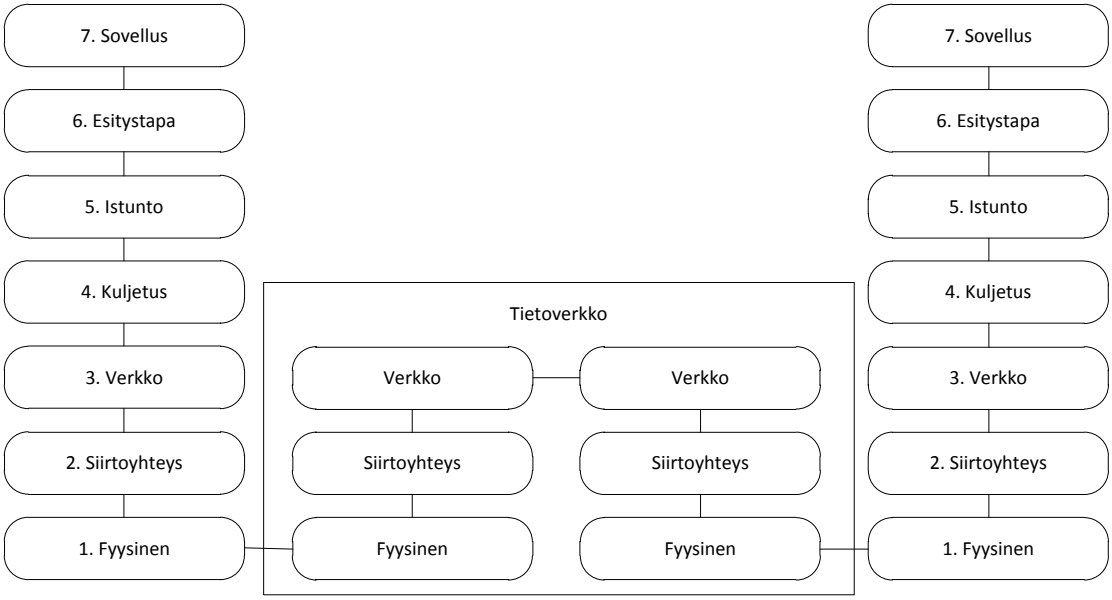
Tietoliikenneverkko on lukuisten protokollien sekä fyysisten laitteiden muodostama laaja kokonaisuus. Verkkotekniikan alkuaikoina lukuisat laitetoimittajat tekivät omat ratkaisunsa tietojärjestelmiensä taustalle ja tuolloin järjestelmän tilaaja oli sidoksissa kyseisen laitetoimittajan tuotteisiin. Tietojärjestelmien yhteensopimattomuus johti lopulta yhtenäiseen käytäntöön, kuinka laitteiden tulisi kommunikoida keskenään, ja näin syntyi ISO:n toimesta käsite OSI-malli. Tätä OSI-mallin käsitettä on nykyään lähes mahdoton sivuuttaa tietoverkkotekniikasta puhuttaessa. (Hämeen-Anttila 2003, 13.)

OSI-malli pyrkii kuvaamaan tietoverkon rakenteet sekä suoritettavat toiminnot kerroksiin jaetun mallin mukaisesti. Tässä mallissa on seitsemän kerrosta ja jokaiselle niistä on määrätty oma roolinsa toimintojen suorittamiseksi. Kerroksiin jakamisen periaatteena on se, että jokaisen kerros palvelee itseään ylempää kerrosta. Kuviossa 1 olevien kerrosten keskustellessa suoraan keskenään, käytetään termiä protolla, kun taas järjestelmän sisällä olevien kerrosten välisestä viestinnästä käytetään nimitystä primitiivi. (Hämeen-Anttila 2003, 14.)

Kerrosten jako ja tehtävät menevät seuraavalla tavalla:

1. Fyysinen kerros suorittaa bittitason toimenpiteet tiedon siirtämisessä järjestelmästä toiseen käyttämällä fyysistä siirtotietä, kuten kuparijohdinta tai valokuitua.
2. Siirtoyhteyskerros huolehtii tiedon luotettavasta siirrosta käyttämällä fyysistä siirtotietä. Kerroksen tehtävä on myös määrittää lähiverkossa käytettävien laitteiden fyysiset osoitteet eli MAC-osoitteet.
3. Verkkokerros huolehtii mm. yhteyden muodostuksesta, ylläpidosta, verkkoliikenteen priorisoinnista sekä yhteyden purkamisesta.
4. Kuljetuskerroksen tehtävä on huolehtia tiedon kuljettamisesta tietoverkon päästä-päähän, pilkkoa kuljetettava tieto datapaketteihin sopivaksi ja vikatilanteissa kerroksen tehtävä on etsiä mm. vaihtoehtoinen reitti kohteeseen.

5. Istuntokerros vastaa mm. salausmenetelmistä, käyttöoikeuksien hallinnoinnista sekä muista tietojärjestelmien suojauksiin liittyvistä toimenpiteistä.
6. Esitystapa määrittää, millaisessa muodossa esimerkiksi asiakkaan ja palvelimen välinen liikenne tulee esittää. Tämä tapahtuu käyttämällä erilaisia koodausjärjestelmiä.
7. Sovelluskerroksen tehtävä on vastata kaikista muista tiedon käsittelyyn liittyvistä toimenpiteistä, joihin alemmat kerrokset eivät osallistu. Tällaisia toimintoja on esimerkiksi verkkosivun esittäminen yksinkertaisessa muodossa. (Hakala & Vainio 2005, 138.)



Kuvio 1. OSI-malli (Hämeen-Anttila 2003, 14.)

2.2 Ethernet

2.2.1 Historiaa

Lähiverkkojen rakentamiseksi on ollut aikojen saatossa useita erilaisia tekniikoita, kuten esimerkiksi ATM ja Frame Relay. Suosituin tapa lähiverkon rakentamiseen on nykyään IEEE 802.3-työryhmän standardisoima Ethernet. Kyseinen tekniikka on vuosien saatossa vallannut suuren jalansijan mm. tuomalla uusia ominaisuuksia ja mahdollistaen yhä suuremmat tiedonsiirtonopeudet.

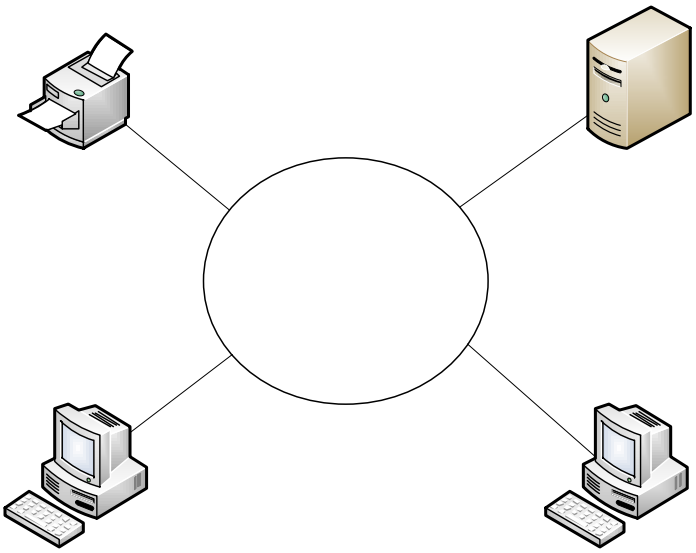
1980-luvulla tietokoneiden yhdistäminen toisiinsa tuli ajankohtaiseksi. Ensimmäisiä uraa uurtavia tekniikoita tähän oli IBM:n kehittämä Token Ring sekä mm. Intelin ja useiden muiden tunnettujen valmistajien yhdessä kehittämä Ethernet. Näissä tekniikoissa kaikki lähiverkon laitteet yhdistettiin loogisessa topologiassa samaan siirtoväylään. Tekniikoiden kompastuskiviä olivat suhteellisen hitaat tiedonsiirtonopeudet sekä menetelmät, joiden perusteella jaettiin päätelaitteille vuorot käyttää siirtotietä. Lukuisten käänteiden jälkeen Ethernet selvisi lopulta voittajaksi muista kilpailevista tekniikoista ja siitä lähtien sitä ovat tukeneet ja kehittäneet useat IEEE-työryhmät. (Jaakohuhta 2005, 22.)

2.2.2 Rakenne

Lähiverkoissa käytetty kaapelointitapa eli topologia määrittää itse verkon rakenteen. Yleisimmät topologiat ovat rengas, väylä sekä tähti. Topologiat jaetaan lisäksi kahteen kategoriaan: fyysinen ja looginen. Fyysinen topologia kuvaa kaapeleiden sekä laitteiden fyysistä sijaintia, kun taas looginen topologia kuvaa verkon rakennetta. (Hämeen-Anttila 2003, 28.)

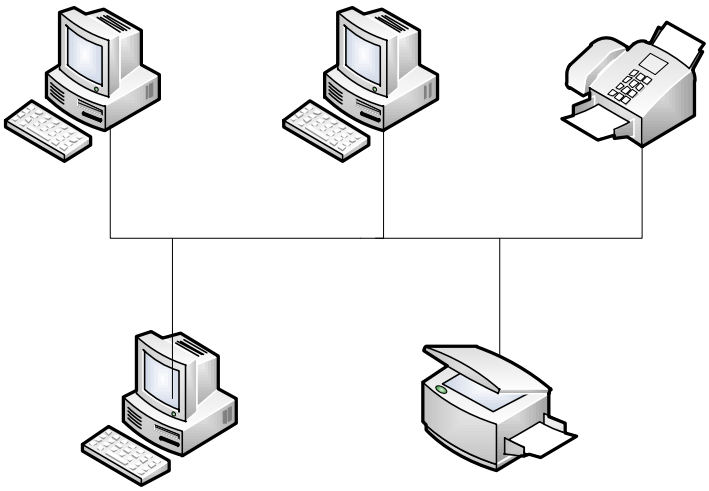
Ensimmäiset tietoverkot rakennettiin käyttämällä koaksiaalikaapelia, mutta nykyään käytäntönä on rakentaa tietoverkko siten, että työasemat sekä tulostimet jne. yhdistetään verkkoon käyttämällä parikaapelia.

Rengasverkossa kaikki laitteet ovat kytkettyinä toisiinsa kuvion 2 mukaisesti. Tieto liikkuu tässä verkossa samansuuntaisesti laitteelta laitteelle ja sen saapumisen ajankohta on laskettavissa. Tällöin puhutaan epädeterministisestä ajoituksesta. Verkon rakenteesta johtuen vaarana on, että tieto saattaisi liikkua ikuisesti verkossa ja lopulta tukkia kaiken siirtoväylän. Ratkaisu on seuraavanlainen: mikäli laite havaitsee saamansa Ethernet-kehyksen itsensä lähettämäksi, se poistaa kyseisen kehyksen verkosta. Kyseinen topologia on kuitenkin hyvin haavoittuvainen, koska yhden laitteen hajoaminen katkaisee viestien välityksen. (Hakala & Vainio 2005, 68.)



Kuvio 2. Rengasverkko

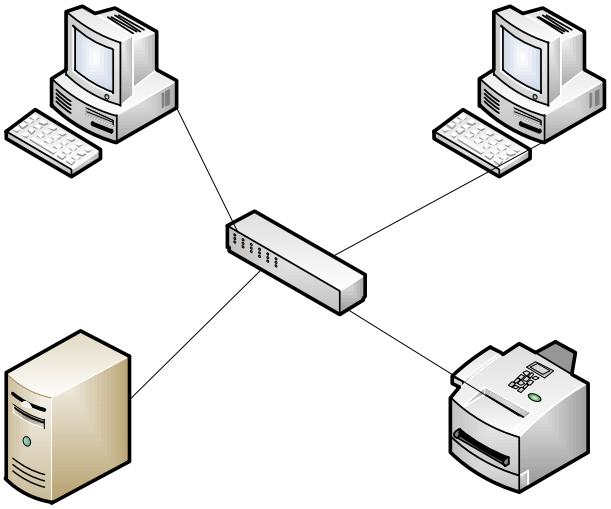
Väyläverkossa kaikki tieto siirtyy samaa siirtoväylää pitkin ja kaikki verkossa olevat laitteet ovat kytkettyinä tähän. Väyläverkossa kehyksille ei ole määritetty mitään tiettyä kulkusuuntaa, vaan kehykset lähetään jokaiseen topologian mahdollistamaan kulkusuuntaan, kuten esimerkiksi kuviossa 3. (Hämeen-Anttila 2003, 69.)



Kuvio 3. Väyläverkko

Tähtiverkon idea perustuu kuvion 4 mukaisesti verkon keskellä olevaan kytkentäpisteeseen, joka on usein kytkin tai muu vastaava verkkolaite. Verkon rakenne mukailee

vanhoja keskustietokoneiden aikaisia ympäristöjä, jossa työasemat yhdistettiin suoraan keskustietokoneeseen. Tämän kytkentätavan etu on helppo hallittavuus sekä vikasietoisuus, sillä yhden tietokoneen hajoaminen ei lamaannuta koko verkon toimintaa. (Hämeen-Anttila 2003, 30.)

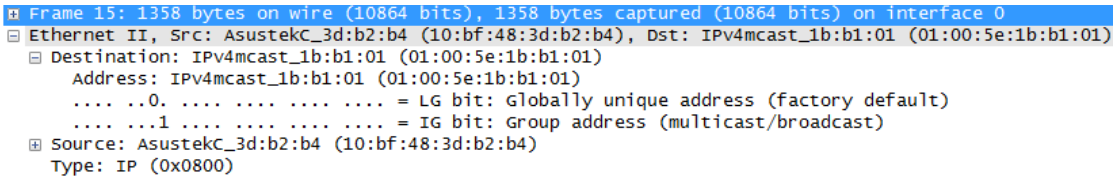


Kuvio 4. Tähtiverkko

2.2.3 Tiedonsiirto

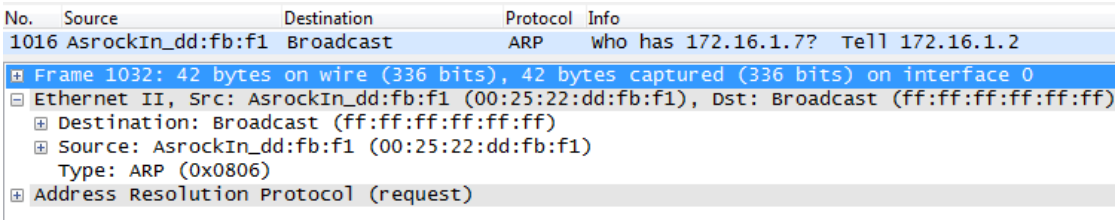
Kaikki Ethernet-verkossa siirrettävä data välitetään OSI-mallin toisella kerroksella kehyksiin (frames) pakattuina. Näitä kehyksiä voidaan lähettää kolmella tavalla ja jokaiselle niistä on oma käyttötarkoituksensa. Nämä lähetystavat ovat unicast, multicast ja broadcast. Broadcast ja multicast ovat Ethernetissä vähemmän käytettyjä lähetysmuotoja, mutta niilläkin on tarkoituksensa. (Jaakohuhta 2005, 83.)

Multicast on yleisesti lähettäjältä useammalle vastaanottajalle tarkoitettu lähetysmuoto. Kyseinen menetelmä säästää verkon siirtokapasiteettia, sillä samaa kehystä ei tarvitse lähettää jokaiselle vastaanottajalle erikseen. Näin tämä välitysmuoto onkin tehokas kuljettamaan esimerkiksi videoneuvotteluissa syntyvää dataa. Multicast-kehyksen tunnistaa hyvin vastaanottajan osoitteen perusteella, kuten kuviossa 5. (Jaakohuhta 2005, 83.)



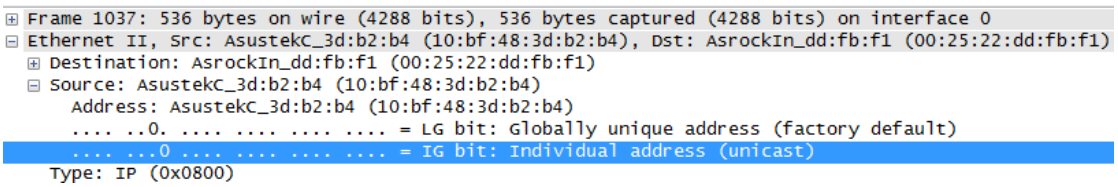
Kuvio 5. Multicast-kehys

Broadcast on tarkoitettu tiedon lähettämiseksi kaikille verkossa oleville laitteille ja sitä käyttävät yleensä hyväksi muut protokollat. Esimeriksi ARP-protokolla käyttää kuviossa 6 broadcastia selvittääkseen IP-osoitteen takana olevan laitteen MAC-osoitteen. Broadcast-kehyksen tunnistaa siitä, että vastaanottajan osoite on heksadesimaalina ilmoitettu muodossa FF:FF:FF:FF:FF:FF. (Jaakohuhta 2005, 84.)



Kuvio 6. ARP-protokollan tekemä kysely

Unicast-kehyksiä käytetään yleisesti tiedon siirtämiseen lähiverkossa. Tämä kehys pitää sisällään kuvion 7 mukaisesti lähettäjän sekä vastaanottajan uniikit kohdeosoitteet. Näistä osoitteista käytetään myös nimitystä MAC-osoite. Saapuneen kehyksen tunnistaminen tapahtuu yksinkertaisesti siten, että laite vertailee kehyksessä olevaa vastaanottajan MAC-osoitetta omaan verkkorajapintansa fyysiseen osoitteeseen. Osoitteen ollessa oikea välitetään kyseinen kehys OSI-mallin mukaisesti ylemmille kerroksille. Mikäli osoite on väärä, jätetään kehys yksinkertaisesti huomioimatta. (Jaakohuhta 2005, 83.)



Kuvio 7. Unicast-kehys

2.2.4 Kehys

Alkuperäinen Ethernet-kehys määritti kaikki OSI-mallin tasolla kaksi suoritettavat toiminnot ja ei siten vastannut täysin IEEE:n määrittämää standardia. Tämä vanhempi kehystyyppi tunnetaan yleensä nimellä Ethernet II. Kyseisessä kehyksessä ei ole mukana vuonohjauksessa tarvittavia tietoja, eikä se myöskään sisällä mitään menetelmiä vastaanotetun tiedon kuittaamiseksi. (Hakala & Vainio 2005, 144.)

Ethernet II -kehyksen ensimmäisenä kenttänä on kuvion 8 mukaisesti ns. tahdistus (preamble), jonka ansiosta kehyksen alkamiskohta voidaan tunnistaa muun liikenteen seasta. Kyseinen kenttä on pituudeltaan vaihteleva, ja se koostuu vuorottelevista 1- sekä 0-biteistä. Tahdistuksen jälkeen kehyksessä on yhden tavun mittainen alkuerotin (Start-of-frame delimiter). Seuraavana ovat vuodossa kuuden tavun mittaiset kohde- ja lähdeosoitteet, eli toisin sanoen MAC-osoitteet. Osoitteiden jälkeen seuraavana on vuorossa kahden tavun mittainen tyyppikenttä. Tämä kenttä kertoo, mille ylemmän tason protokollalle kehyksessä oleva hyötykuorma (payload) tulee ohjata. Tyyppikentän jälkeen tulee varsinainen hyötykuorma, jonka pituus vaihtelee 46:n ja 1500 tavun välillä. Kehyksen viimeisenä ja päättävänä kenttänä on neljän tavun mittainen varmistuskenttä, joka sisältää otsikkotiedoista sekä hyötykuormasta lasketun varmistussumman (FSC). (Hakala & Vainio 2005, 144; Hämeen-Anttila 2003, 33-34.)

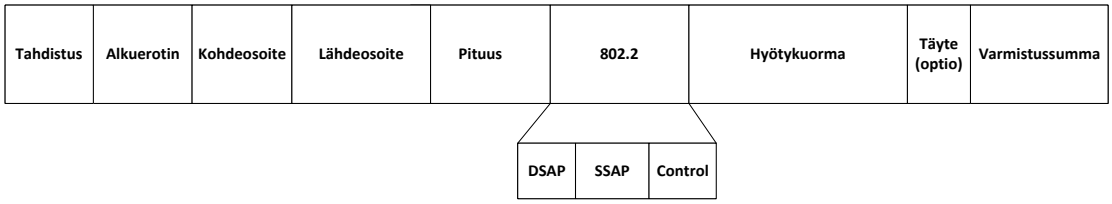
Tahdistus	Alkuerotin	Kohdeosoite	Lähdeosoite	Tyyppi	Hyötykuorma	Varmistussumma

Kuvio 8. Ethernet II-kehys

Standardi 802.2 määrittää väylänvarauksesta riippumattomille ja kaikille IEEE:n määrittelemille kehyksille yhteiset käytännöt otsikkotietojen sijoittamisesta. Nämä myös loogiseksi siirtoyhteydeksi (LLC) kutsutut määritykset sijoitetaan kehyksessä hyötykuorman alkuun. LLC ottaa vastaan sekä sisällyttää ylemmillä tasoilla olevien protokollien tiedot, ja tuolloin kehys voidaan määrittää informaatio-, ohjaus- tai kuittauskehykseksi. (ANSI/IEEE Std 802.2 Part 2: Logical Link Control 1998, 39-47.)

IEEE 802.2-standardin määritelmä kehys sisältää kolme otsikkokenttää: DSAP, SSAP sekä Control. DSAP- ja SSAP-kentät sisältävät tiedon käytössä olevasta ylemmän tason protokollasta. Control-kenttä määrittää, onko kehys tarkoitettu kuittaamiseen, ohjaamiseen vai sisältääkö se siirrettävää hyötykuormaa. LLC-kerros muodostaa näistä otsikkokentistä ns. kapselin, joka sijoitetaan standardin 802.3 mukaiseen kehykseen. (IEEE Std 802.3-2012 2012, 28-29.)

IEEE 802.3-kehyksen rakenne on kuviossa 9 sama kuin edellä mainitun vanhemman Ethernet II:n, mutta tyyppikentän sijaan se sisältää kentän, joka ilmoittaa hyötykuorman pituuden tavuina. Mikäli kyseinen kehys ei sisällä ollenkaan hyötykuormaa tai kehyksen pituus jäisi alle minimipituuden, lisätään em. LLC-kapselin perään riittävä määrä täytemerkkejä (padding), että vaadittava 46 tavun minimipituus täyttyisi. (Hakala & Vainio 2005, 145; Jaakohuhta 2005, 86; IEEE Std 802.3-2012 2012, 53.)



Kuvio 9. IEEE 802.3-kehys

2.2.5 Jumbo frame

Määritysten mukaan maksimi koko Ethernet-kehyksessä siirrettävälle hyötykuormalle on 1500 tavua. Nykyään tiedonsiirtonopeudet ja yhteydet ovat jo sitä luokkaa, että ne mahdollistavat suurienkin hyötykuormien kuljettamisen yhdessä kehyksessä. IEEE ei ole vielä tehnyt standardia, joka mahdollistaisi suurempien hyötykuormien kuljettamisen.

Ethernet Alliance on yhdistynyt liitto, joka on ottanut tehtäväksi tukea ja kehittää Ethernet-tekniikkaa. Kyseinen liitto on lukuisten valistajien kanssa sopinut käytännöistä, joilla yhteen kehykseen saataisiin enemmän hyötykuormaa. Hyötykuorman kasvaessa, kasvaa myös kehyksen koko. Tästä johtuen kyseisistä kehyksistä käytetään nimitystä *Jumbo frame*. (Ethernet Jumbo Frames 2009, 2).

Kyseisestä tekniikasta hyötyvät eniten palvelin- ja laitesalit, jossa siirrettävää tietoa on paljon. Todellinen tiedonvälityskapasiteetti kasvaa ja aktiivilaitteiden käyttöaste laskee, koska käsiteltäviä kehyksiä on vähemmän. Joissain tapauksissa voidaan käyttää kehystä, joka pystyy kuljettamaan jopa 9000 tavua. Kyseiseen kehyskokoon pystyviä protokollia on muun muassa NFS ja iSCSI. (Ethernet Jumbo Frames 2009, 5.)

Tällä on kuitenkin omat haittavaikuksensa. Käytettäessä suuremaa kehyskokoa, kasvaa myös tiedonsiirtoviiveet. Tämä saattaa olla haitallista pientä viivettä vaativissa sovelluksissa, joita ovat esimerkiksi videoneuvottelut. Myös kaikkien samassa lähiverkossa olevien laitteiden pitää tukea kyseistä ominaisuutta. Niinpä suurimmat käyttösovellukset ovat tällä hetkellä lähiverkosta eristetyt virtuaalisointi- ja levyjärjestelmät. (Ethernet Jumbo Frames 2009, 5-7.)

2.2.6 CSMA/CD

Tietoverkon topologiasta riippumatta ja fyysisten rajoitusten vuoksi Ethernet-verkossa vain yksi laite saa lähettää kehyksensä kerrallaan. Syynä tähän on fyysisessä kaapelissa kulkevan sähköisen signaalin sekoittuminen tunnistamattomaksi sekameteliksi, mikäli useampi laite yrittäisi viestiä samanaikaisesti. Lähettävä laite varaa siirtotien käyttöönsä standardin mukaisesti koko lähetystapahtuman ajan ja tästä käytetään termiä väylänvaraus (MAC).

Ethernetissä ei jaeta lähetysvuoroja laitekohtaisesti, vaan verkkoon liitetyt laitteet joutuvat kilpailemaan keskenään käytettävästä siirtoväylästä. Lähettääkseen tietoa laitteen pitää varmistaa siirtoväylän käytettävyys. Tämä tapahtuu tarkkailemalla fyysisen siirtotien jännitetasoa. Mikäli jännitettä ei havaita on siirtotie vapaa käytettäväksi ja tarvittavat kehykset voidaan lähettää.

Usein tulee kuitenkin vastaan tilanne, jossa useampi verkkoa käyttävä laite on havainnut siirtotien vapaaksi samanaikaisesti ja tuolloin lähettivät kehyksensä. Kyseisessä tilanteessa siirtotielle lähetetyt kehykset törmäävät (collision) toisiinsa ja aiheuttavat jännitetason nousun. Laitteet havaitsevat (detection) törmäyksen perustuen tähän jännitetason nousuun ja ensimmäisenä törmäyksen havainnut laite lähettää siitä tiedon ns. ruuhkasignaalin avulla. Törmäykseen joutuneet kehykset tuhoutuvat

ja ne joudutaan lähettämään uudestaan. Jotta uudelleenlähetys ei johtaisi taas uuteen törmäykseen, odottavat laitteet standardin mukaisesti määritetyn ajan ennen uudelleenlähetystä. (Hakala & Vainio 2005, 75.)

3 IEEE 802.1X

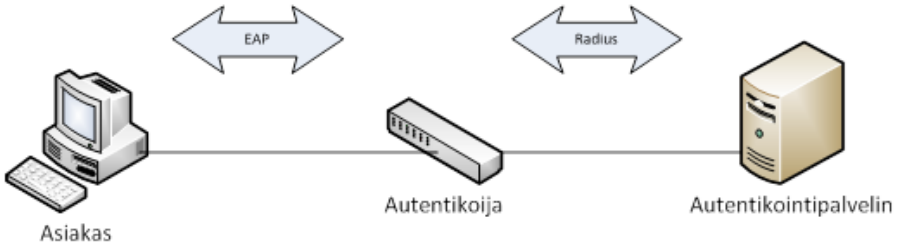
3.1 Kuvaus

IEEE 802.1X on nimensä mukaisesti IEEE:n määrittelemä standardi, jota voidaan käyttää käyttäjien sekä päätelaitteiden todennukseen niin langallisissa kuin langattomissa lähiverkoissa. 802.1X mahdollistaa EAP-metodien kuljettamisen verkkolaitteiden välillä sekä verkoista toiseen, koska EAP-viestit paketoidaan EAPOL-paketiksi. (Aboba ym. 2003, 2. & Port-Based Network Access Control 2010, 2.)

802.1X:n parhaita ominaisuuksia on sen soveltuvuus EAP:n ohella monenlaiseen verkkoympäristöön ja sovellukseen. Yksi käytetyimmistä sovelluksista lähiverkossa on porttikohtainen todennus, jossa esimerkiksi yrityksen tietoverkkoon pyrkivä henkilö tai päätelaite varmennetaan ennen verkkoon pääsyä. (Aboba ym. 2003, 2; Port-Based Network Access Control 2010, 2.)

802.1X:n toiminta perustuu kuvion 10 mukaisesti kolmeen osapuoleen: asiakkaaseen, autentikoijaan sekä autentikointipalvelimeen. Asiakasta kuvaa joko päätelaite tai henkilö. Autentikoijan roolia hoitaa yleensä kytkin, joka samalla toimii ns. välikätenä asiakkaan sekä autentikointipalvelimen välillä. Itse autentikointipalvelin taas sisältää tietokannan käyttäjien tai päätelaitteiden tunnistetiedoista sekä niille määritellyistä oikeuksista. Standardi ei kuitenkaan pakota käyttämään erillistä autentikointipalvelinta, vaan kyseistä roolia voi tarvittaessa hoitaa myös todennuksesta vastaava verkkolaite. (Aboba ym. 2003, 2; Port-Based Network Access Control 2010, 33.)

Autentikoija sijaitsee aina asiakkaan ja palvelimen välissä, koska kumpikin osapuoli käyttää eri protokollaa viestien vaihtamiseen. Asiakas ja autentikoija vaihtavat keskenään EAP-viestejä, kun taas autentikoija sekä palvelin käyttävät RADIUS-protokollaa. (Port-Based Network Access Control 2010, 48.)

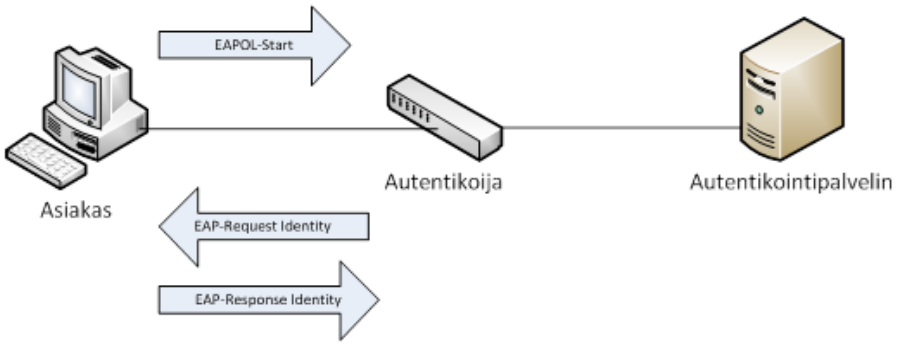


Kuvio 10. IEEE 802.1X:n osapuolet

3.2 Todennusprosessi

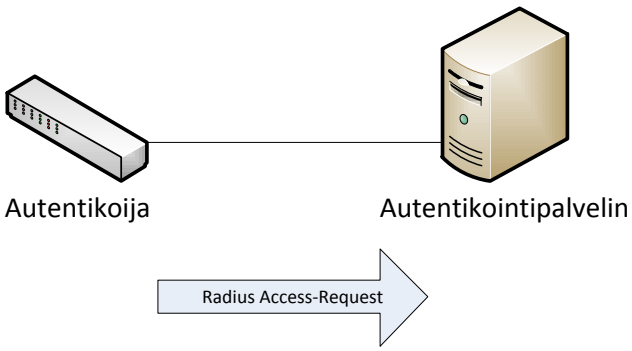
Kuvion 11 alkutilanteessa asiakkaan päätelaite kytkeytyy fyysisesti lähiverkkoon ja kytkimessä oleva verkkorajapinta on vielä kiinni. Mikäli päätelaitteessa on määritetty porttikohtainen todennus, se lähettää kytkimelle viestin EPOL-Start. Kytkin huomaa portissa liikehdintää ja tiedustelee verkkoon kytkeytyneeltä laitteelta tietoja. Tiedustelu tapahtuu lähettämällä viesti EAP-Request Identity. Kytkin tekee tämän toimenpiteen vaikka EAPOL-Start -viestiä ei tulisikaan. (Port-Based Network Access Control 2010, 51.)

Päätelaite vastaa tähän viestiin lähettämällä EAP-Response Identity -viestin, mikä sisältää asiakkaan tunnistamiseen vaadittavat tiedot, kuten esimerkiksi työaseman nimen. Mikäli kytkin ei saa vastausta EAP-Request Identity -viestiin, se odottaa määritellyn ajan ennen uudelleenyritystä. Mikäli useammasta yrityksestä huolimatta päätelaite ei vastaa, kytkin sulkee käytössä olevan portin liikenteeltä. Usein nämä kytkimeen määritetyt asetukset, kuten odotusaika sekä uusintayritysten määrä ovat laite- ja valmistajakohtaisia.



Kuvio 11. Todennuksen ensimmäinen vaihe

Kun asiakkaan päätelaite vastaa kyselyyn, siirtyy todennus toiseen vaiheeseen. Kytkin etsii IP-osoitteen perusteella sille määritellyn autentikointipalvelimen. Mikäli palvelin ei vastaa, yrittää kytkin asetuksissa määritellyn ajan uudelleen tai siirtyy mahdollisesti käyttämään toissijaista autentikointipalvelinta. Ennen tietojen välittämistä palvelimelle, kytkin purkaa asiakkaalta saadun EAP-viestin, paketoi siitä saamansa tiedot RADIUS-Access-Request -viestiin sekä lähettää kyseisen viestin palvelimelle kuvion 12 mukaisesti. Käytettävän protokollan ei ole pakko olla RADIUS, mutta kyseinen menetelmä on muodostunut lähes de facto -standardiksi. Lisäksi tässä opinnäytetyössä käytetty protokolla oli RADIUS. (Port-Based Network Access Control 2010, 48; Rigney ym. 2000, 6.)

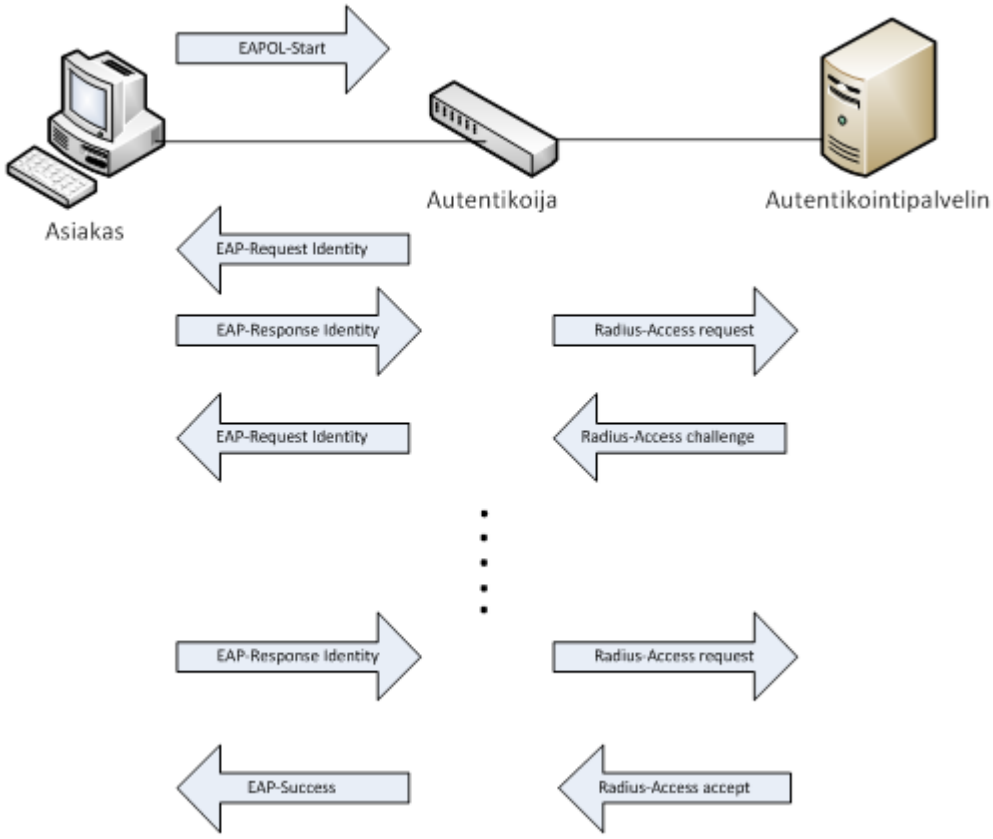


Kuvio 12. Todennuksen toinen vaihe

Kolmannessa vaiheessa autentikointipalvelin saa kytkimeltä RADIUS-Access-Request -viestin ja asiakkaan tiedoista riippuen palvelin joko hyväksyy pyynnön tai hylkää sen. Jos palvelin päättää hylätä kytkimeltä saapuvan pyynnön, vastaa se tälle lähettämällä

viestin RADIUS-Access-Reject. Kytkin ilmoittaa tästä asiakkaalle viestillä EAP-Failure. Yleisimmät syyt hylkäämiseen ovat asiakkaan virheelliset tunnistustiedot ja palvelimen sekä kytkimen välisen tunnussanan eroavaisuus. Toinen mahdollinen syy on se, että palvelinta ei ole määritetty vastaamaan kyseisen kytkimen pyyntöihin. (Rigney ym. 2000, 6; Aboba ym. 2004, 22-23.)

Teoriassa autentikointipalvelin voisi päättää asiakkaan todentamisen jo tähän vaiheeseen, lähettämällä kytkimelle RADIUS-Access-Accept -viestin. Tämä ei kuitenkaan ole suotavaa ja niinpä palvelin lähettää todennusta suorittavalle kytkimelle RADIUS-Access-Challenge -viestin. Kyseinen viesti sisältää todennuksessa käytettävät määritykset, kuten esimerkiksi käytettävän EAP-metodin. Saadessaan tämän viestin kytkin välittää sen asiakkaalle muodossa EAP-Request Identity ja asiakas vastaa tähän lähettämällä viestin EAP-Response Identity. Määrityksistä riippuen tätä haaste- ja vasteviestintää saattaa jatkua pitkäänkin, kunnes palvelimella on riittävän hyvä käsitys asiakkaasta. Tuolloin palvelin voi hyväksyä kytkimen pyynnön asiakkaan todentamiseksi. Tämä tapahtuu kuvion 13 mukaisesti lähettämällä kytkimelle RADIUS-Access-Accept -viesti. Kytkin vielä ilmoittaa asiakkaalle onnistuneesta todennuksesta lähettämällä tälle viestin EAP-Success. (Rigney ym. 2000, 7; Aboba ym. 2004, 22-23; Port-Based Network Access Control 2010, 51.)



Kuvio 13. Asiakkaan onnistunut todennus

3.3 EAP

EAP (Extensible Authentication Protocol) on todennuksessa käytettävä protokolla ja se sekoitetaan usein itse todennusprotokollaksi. Kuitenkin se on vain IEFT-organisaation määrittelemä standardi, kuinka todennuksessa käytettävät viestit tulisi kuljettaa verkko- ja päätelaitteen sekä autentikointipalvelimen välillä. EAP suunniteltiin käytettäväksi verkkopohjaisessa todennuksessa, kun IP-tason yhteyttä ei ole saatavilla ja protokollan tarkoitus on tarjota luotettava tiedonsiirto todennuksen aikana. (Aboba ym. 2004, 5.)

EAP tukee useita todennusmetodeita ja näillä on myös erilaiset vahvuudet muun muassa salauksessa sekä turvaominaisuudet, joiden perusteella voidaan valita sopiva EAP-metodi. EAP ei tarvitse IP-osoitetta välittääkseen viestejä, koska protokolla toimii kappaleessa 2.1 mainitun OSI-mallin toisella kerroksella. Näin tietoverkkoon pyrkivä päätelaite tai henkilö voidaan turvallisesti todentaa ennen IP-osoitteen määrittämistä sekä tietoverkkoon pääsyä. (Aboba ym. 2004, 5.)

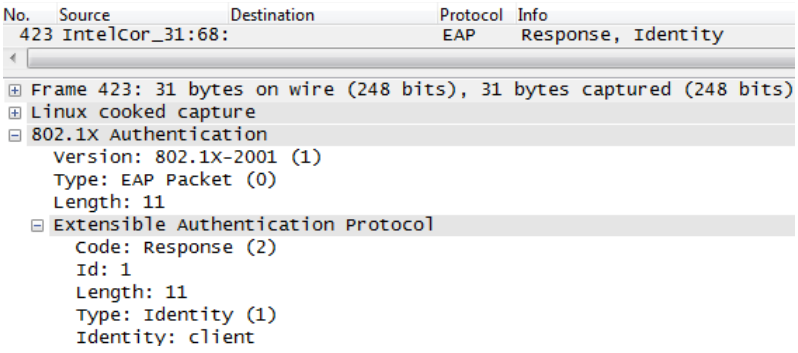
Tietoturvan kannalta EAP antaa mahdollisuuden hallita paremmin käyttäjien oikeuksia verkkolaitteisiin sekä pääsyä itse tietoverkkoon. Todennusta pyytävän aktiivilaitteen ei tarvitse tietää mitään käytettävästä todennustavasta, sillä se vain välittää EAP-viestejä autentikointipalvelimen, käyttäjän ja työaseman välillä. Kaikkien edellä mainittujen hyötyjen saavuttamiseksi tarvitaan tietysti hieman tietotaitoa sekä kunnollista suunnitelmaa tietoverkon toiminnan osalta. Suunnitelman tulisi vastata kysymyksiin, miten ja milloin todennusta tarvitaan sekä kuinka toimitaan mahdollisessa vikatilanteessa. (Aboba ym. 2004, 7.)

EAP-kehyksiä on määritelty taulukon 1 mukaisesti neljä kappaletta ja ne ovat toiminnaltaan hyvin selkeitä. Kehyksistä kolme on autentikoijan käytössä, joilla muun muassa kysellään asiakkaalta tietoja sekä vastataan todennusprosessin lopputulokseen.

Taulukko 1. EAP-kehyksien koodit

Koodi	**Kuvaus**
1	Request
2	Response
3	Success
4	Failure

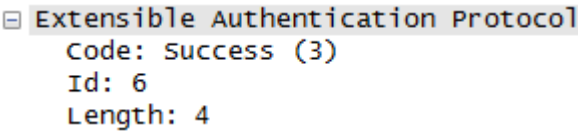
Asiakas käyttää edellä mainitun taulukon 1 kehyksistä ainoastaan yhtä kappaletta, joka on EAP-Response. Tämä tapahtuu esimerkiksi vastatessa todennuksesta huolehtivan verkkolaiteen pyyntöihin, kuten kuvion 14 Wireshark kuvankaappaus havainnollistaa.



Kuvio 14. EAP-Response Identity

Mikäli osapuolet havaitsevat jonkin muun EAP-kehyksen, tulee se standardin mukaan jättää huomioimatta. Kehyksen tärkein kenttä on datakenttä, mikä sisältää kuljetettavan datan asiakkaan ja autentikointipalvelimen välillä. Loput neljästä kentästä kertovat kehyksen tyypin, tunnisteen ja pituuden. (Aboba ym. 2004, 19.)

Autentikoijan asiakkaalle lähettämä Request-kehys tunnistetaan koodista 1 ja asiakkaan vastaus tähän kehyksellä Response taas koodista 2. Mikäli autentikoija saa hyväksytyn vastauksen autentikointipalvelimelta, välittää se tiedon asiakkaalle kehyksellä Success, joka tunnistetaan koodista 3. Muutoin asiakas saa koodilla 4 merkityn Failure-kehyksen, joka tarkoittaa samalla todennuksen epäonnistumista. Success ja Failure eivät sisällä mitään hyötydataa, kuten kuvion 15 Wireshark kuvankaappauksesta voidaan todeta. (Aboba ym. 2004, 20, 22-23.)



Kuvio 15. EAP-Success

3.3.1 EAP-MD5 CHAP

Message Digest 5 Challenge Handshake Authentication Protocol on yksi varhaisimmista todennusprotokollista ja yhdistettynä EAP-metodin kanssa se on varsin yksinkertainen menetelmä todennuksen suorittamiseksi. CHAP on määritelty vuonna 1996 käytettäväksi PPP-yhteyksien kanssa, mutta protokollan toiminta on samanlainen

tässäkin tapauksessa. Tämän EAP-metodin toiminta perustuu ns. tunnussanaan sekä MD5-haasteeseen, joka muodostetaan käyttämällä hyväksi osapuolten tiedossa olevaa tunnussanaa sekä MD5-algoritmia. (Simpson 1996, 1.)

MD5-algoritmi kehitettiin korvaamaan MD4:n ja aikaisempien versioiden mahdollisia heikkouksia. Algoritmi tuottaa 128-bittisen tiivisteen ja laskennallisesti on ajateltu olevan täysi mahdottomuus tuottaa kaksi tai useampi arvoltaan täysin samanlainen tiiviste, erilaisella syötteellä. Algoritmin idea perustuu siihen, että itse tuotetusta tiivisteestä ei pystytä mitenkään päättelemään algoritmin läpi ajettua syötettä, kuten esimerkiksi salasanaa. Muita MD5-algoritmia hyödyntäviä osa-alueita ovat mm. datan eheyden varmistaminen tiedostonsiirrossa sekä vapaiden ohjelmistojen aitouden varmistaminen. (Rivest 1992, 5.)

Tämän EAP-metodin suurin heikkous on se, että käytössä oleva tunnussana säilytetään selkokielisenä osapuolten kesken. Kuitenkaan itse tunnussanaa ei ikinä siirretä tietoverkossa, vaan osapuolet lähettävät keskenään MD5-algoritmin ja tunnussanan perusteella laskettuja tiivisteitä ja vertailemalla näitä tiivisteitä omiin laskutoimituksiin tehdään päätös toisen osapuolen oikeudellisuudesta. (Blunk & Vollbrecht 1998, 10.)

Kyseistä metodia voi suositella käytettäväksi vain sellaisissa tapauksissa, jossa voidaan olla täysin varmoja siitä, että todennuksessa olevien osapuolten viestejä ei pystytä kaappaamaan, koska osapuolten lähettämät haaste- ja vasteviestit kulkevat tietoverkossa salaamattomia. Kaappaamalla näitä viestejä, voidaan nykypäivän laskentateholla, sanakirjoihin perustuvilla hyökkäysmenetelmillä sekä tietämällä viestien muodostamisessa käytetyn algoritmin rakenne, murtaa käytössä oleva salasana. (Tomes & Baggett 2011.)

3.3.2 EAP-MS-CHAPv2

Microsoft Challenge-Handshake Authentication Protocol mahdollistaa molempien osapuolien todentamisen. Protokolla kuljettaa mukanaan asiakkaan haastetta Response-viestissä ja autentikoijan vastausta Success-viestissä. Kyseinen protokolla vastaa toiminnaltaan aikaisemmin esiteltyä CHAP-protokollaa ja se onkin vain Microsof-

tin variaatio kyseisestä protokollasta. Tämän EAP-metodin etuna on se, että käytettäviä tunnistetietoja ei tarvitse säilyttää selväkielisenä autentikointipalvelimella ja lisäksi se mahdollistaa mm. vahvempien salausalgoritmien käytön viestien suojaamiseksi. Lisäsuojaa tuo myös se, että viestien vastaanotossa ja lähetyksessä käytetään eri salausavaimia. Kyseinen EAP-metodi mahdollistaa myös asiakkaan salasanan vaihtamisen. (MS-CHAP v2 2014; Kamath & Palekar 2004, 2.)

Asiakkaan liittyessä verkkoon pyytää verkkolaite tältä tietoja lähettämällä EAP-Request Identity -viestin. Asiakas vastaa tähän viestillä EAP-Response Identity, joka samalla sisältää tiedon asiakkaan userID:stä. Tämän jälkeen todennusta vaativa verkkolaite lähettää todennuspyynnön autentikointipalvelimelle. Palvelin vastaa lähettämällä asiakkaalle MS-CHAPv2 Challenge -viestin, joka sisältää istunnon tunnisteen ja satunnaisesti muodostetun haastetunnisteen. (MS-CHAP v2 2014; Potter & Zamick 2002, 2.)

Asiakas vastaa tähän haasteeseen lähettämällä EAP-MS-CHAPv2 Response -viestin. Kyseinen viesti muodostetaan asiakkaan nimen ja satunnaisesti muodostetun haastetunnisteen perusteella. Lisäksi kyseisen viestin muodostamiseksi otetaan vielä mukaan palvelimelta saatu haastetunniste, istunnon tunniste sekä MD4-algoritmilla muodostettu tiiviste asiakkaan salasanasta. (MS-CHAP v2; Kamath & Palekar 2004, 7-8.)

Autentikointipalvelin tarkistaa asiakaan lähettämät tiedot vertaamalla saamaansa vastausta laskemaansa arvoon, minkä jälkeen lähettää vastauksena MS-CHAPv2-Response -viestin. Viesti sisältää tiedon onnistuneesta tai epäonnistuneesta yhteydenmuodostuksesta. Tämän perusteella asiakas päättää käyttää yhteyttä lähettämällä viestin EAP-MS-CHAPv2 Success Response tai sitten se lopettaa yhteyden muodostamisen. (MS-CHAP v2; Kamath & Palekar 2004, 8-9.)

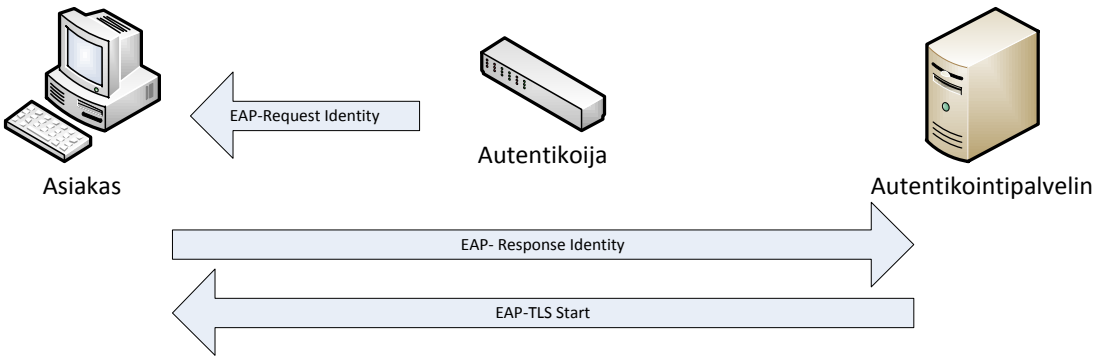
EAP toimii tässä metodissa vain viestien kuljettajana asiakkaan sekä verkkolaiteen välillä. Verkkolaite tulkitsee palvelimen vastaukset välittäessään viestejä ja tekee sen perusteella päätöksen verkkorajapinnan avaamisesta asiakkaalle.

3.3.3 EAP-TLS

EAP-TLS (Transport Layer Security) on EAP-metodi, joka mahdollistaa todennuksessa osallisena olevien osapuolien molemminpuolisen todennuksen, suojatun salausmenetelmän valitsemisen sekä salausavaimien vaihdon osapuolten välillä. (Adoba ym. 2008, 1-2.)

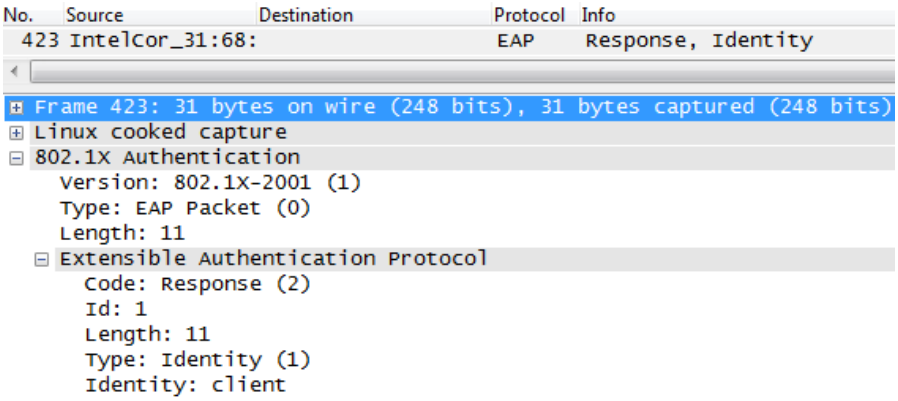
Yksi kyseessä olevan metodin vahvuuksista on varmenteisiin perustuva osapuolten välinen todennus, mikä mahdollistaa vaivattoman käyttökokemuksen loppukäyttäjälle. Juuri tämä ominaisuus oli vahva valintakriteeri lopullisen työn suorittamiseksi. Koska tässä opinnäytetyössä käytetty metodi on EAP-TLS, otetaan se seuraavaksi tarkempaan käsittelyyn. Kaikki metodia todentavat kuvankaappaukset on otettu Wireshark-ohjelmalla ja verkkoympäristönä on juuri tätä tarkoitusta varten rakennettu testiverkko. Lisäksi vaiheittain suoritettavia toimenpiteitä kuvataan vain loogisesti osapuolten välillä ja EAP-metodin näkökulmasta. Välissä olevan verkkolaitteen toimenpiteisiin ei oteta todennuksen aikana kantaa.

Kuviossa 16 on esitetty EAP-TLS -metodin ensimmäinen vaihe. Asiakkaan, eli tässä tapauksessa työaseman liittyessä tietoverkkoon, lähettää todennusta vaativa verkkolaite pyynnön tunnistetiedoista viestillä EAP-Request Identity.



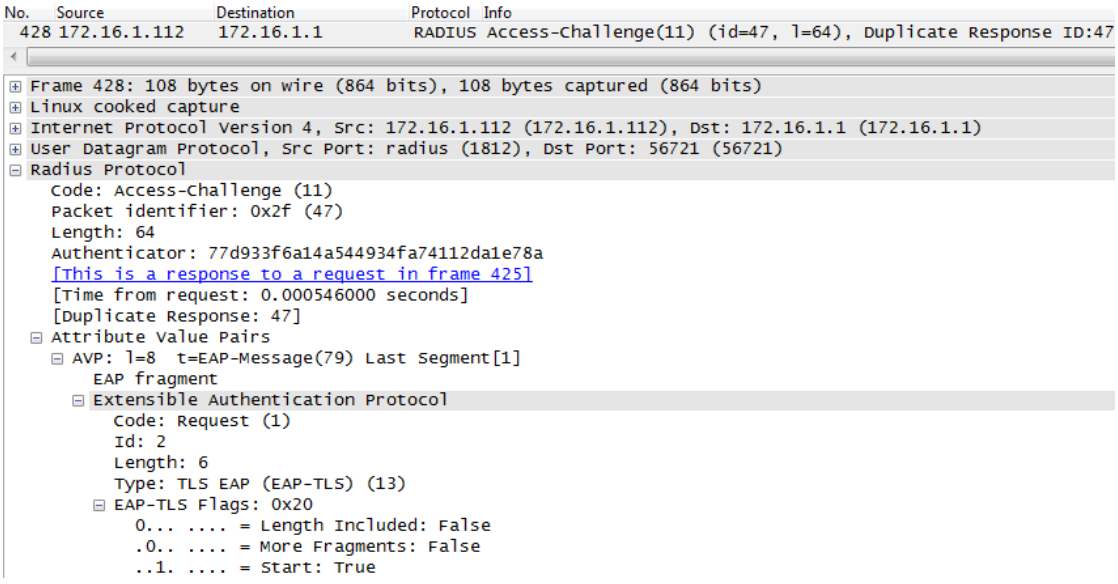
Kuvio 16. EAP-TLS ensimmäinen vaihe

Asiakkaan lähettämä viesti sisältää vastauksen verkkolaitteen tekemään kyselyyn todennettavan tunnistetiedoista, joka on yleensä työaseman tai käyttäjän nimi. Kyseinen viesti on todennettu kuviossa 17.



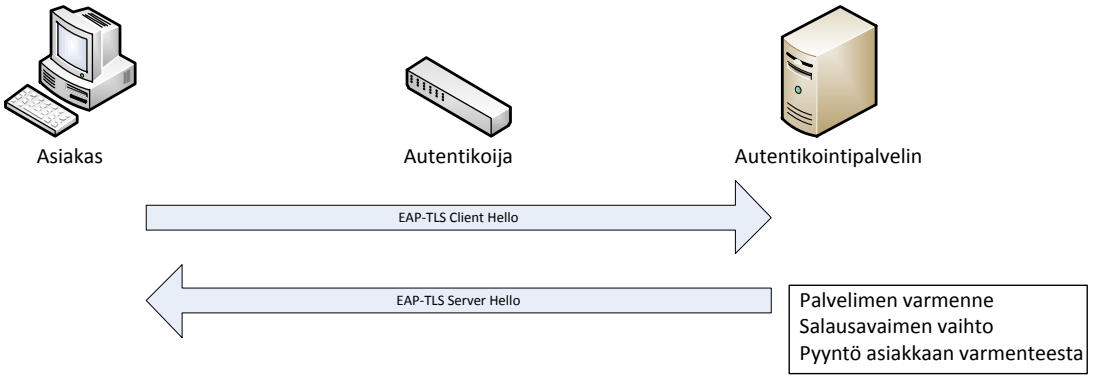
Kuvio 17. EAP-Response Identity

Varsinainen EAP-TLS -yhteyden muodostaminen alkaa kuitenkin autentikointipalvelimen saadessa asiakkaan tunnistetiedot. Palvelin vastaa tähän lähettämällä asiakkaalle haasteen kuviossa 18. Kyseinen haaste on kapseloitu RADIUS-protokollan sisälle attribuuttipariksi ja sen tyyppinä on EAP-TLS. Tämän viestin tärkein kohta on Start-bitin arvo, joka on tässä viestissä 1. Kyseinen parametri kertoo asiakkaalle sen, että palvelin on valmis aloittamaan TLS-yhteyden muodostuksen. (Adoba ym. 2008, 3.)



Kuvio 18. EAP-TLS Start

Saadessaan palvelimelta kehotuksen aloittaa kättely, siirtyy EAP-TLS toiseen vaiheeseen kuvion 19 mukaisesti.



Kuvio 19. EAP-TLS toinen vaihe

Tässä vaiheessa asiakas lähettää viestin EAP-TLS Client Hello. Kyseinen viesti sisältää kuvion 20 mukaisesti tiedot asiakkaan tukemasta TLS-versiosta, istunnon tunnisteen, satunnaisluvun ja listauksen asiakkaan tukemista salausmenetelmistä. (Adoba ym. 2008, 3.)

```
No.   Source              Destination         Protocol  Info
  429 IntelCor_31:68:                         TLSv1   Client Hello

⊞ Frame 429: 125 bytes on wire (1000 bits), 125 bytes captured (1000 bits)
⊞ Linux cooked capture
⊟ 802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: EAP Packet (0)
    Length: 105
  ⊟ Extensible Authentication Protocol
      Code: Response (2)
      Id: 2
      Length: 105
      Type: TLS EAP (EAP-TLS) (13)
    ⊞ EAP-TLS Flags: 0x80
      EAP-TLS Length: 95
    ⊟ Secure Sockets Layer
      ⊟ TLSv1 Record Layer: Handshake Protocol: Client Hello
          Content Type: Handshake (22)
          Version: TLS 1.0 (0x0301)
          Length: 90
        ⊟ Handshake Protocol: Client Hello
            Handshake Type: Client Hello (1)
            Length: 86
            Version: TLS 1.0 (0x0301)
          ⊟ Random
              gmt_unix_time: Apr 22, 2014 02:57:59.000000000 FLE Daylight Time
              random_bytes: a89350ef005b4d8df7691141b97a0f98075ec62d27647422...
            Session ID Length: 0
            Cipher Suites Length: 24
          ⊞ Cipher Suites (12 suites)
            Compression Methods Length: 1
          ⊞ Compression Methods (1 method)
            Extensions Length: 21
          ⊞ Extension: renegotiation_info
          ⊞ Extension: elliptic_curves
          ⊞ Extension: ec_point_formats
```

Kuvio 20. EAP-TLS Client Hello

Asiakkaan tukemat salausmenetelmät ovat tarkemmin esiteltynä kuviossa 21 ja niitä on tässä tapauksessa kaksitoista kappaletta.

```
⊟ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 86
    Version: TLS 1.0 (0x0301)
  ⊞ Random
    Session ID Length: 0
    Cipher Suites Length: 24
  ⊟ Cipher Suites (12 suites)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
      Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
      Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
      Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
      Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
      Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
      Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
    Compression Methods Length: 1
```

Kuvio 21. Asiakkaan tukemat salausmenetelmät

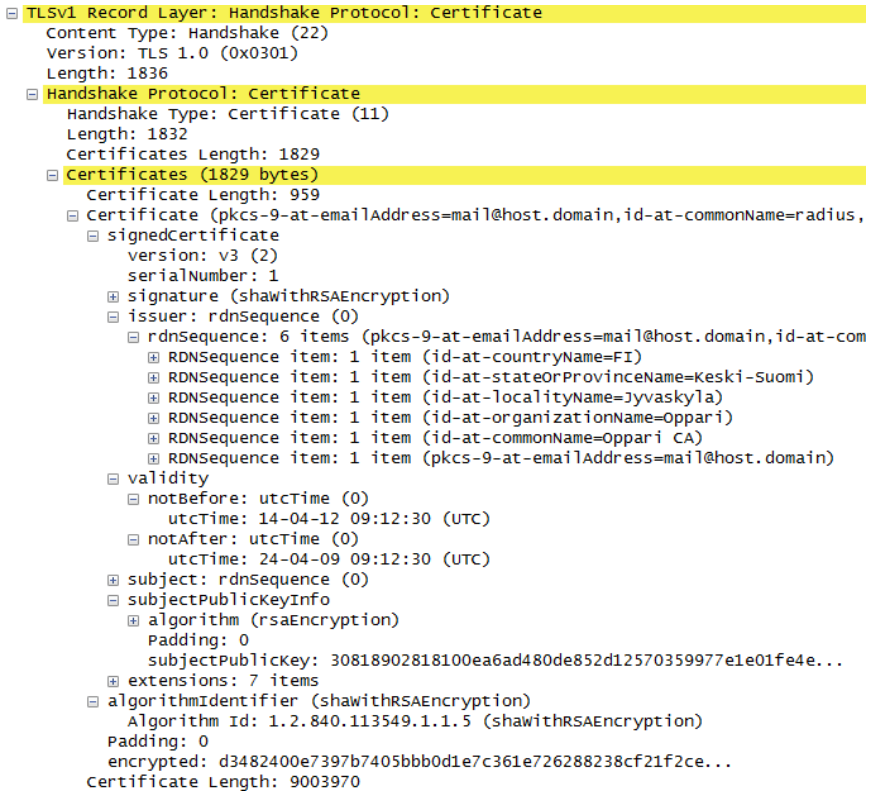
Palvelin vastaa asiakkaan kättelypyyntöön lähettämällä viestin EAP-TLS Server Hello. Tämä viesti sisältää mm. palvelimen varmenteen, julkisen salausavaimen, käytettävän salausmenetelmän sekä istunnon tunnisteen. Kuviossa 22 on todennettu tämä palvelimen lähettämä Server Hello -viesti. (Adoba ym. 2008, 4.)

```
⊟ Secure Sockets Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 49
    ⊟ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 45
        Version: TLS 1.0 (0x0301)
      ⊟ Random
          gmt_unix_time: Apr 22, 2014 02:58:01.000000000 FLE Daylight Time
          random_bytes: 63c2bee54dc774c42ce05c696cd33c742baa31c52334f557...
        Session ID Length: 0
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 5
      ⊞ Extension: renegotiation_info
  ⊟ TLSv1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 1836
    ⊟ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1832
        Certificates Length: 1829
      ⊟ Certificates (1829 bytes)
          Certificate Length: 959
        ⊞ Certificate (pkcs-9-at-emailAddress=mail@host.domain,id-at-commonName=radius,id-at-organizationName=Oppari,
          Certificate Length: 9003970
```

Kuvio 22. EAP-TLS Server Hello

Palvelimen lähettämä varmenne ja sen myöntäjä on esitelty tarkemmin kuvion 23 Wireshark kuvankaappauksessa. Lisäksi kuviossa näkyy myös käytettävän varmen-

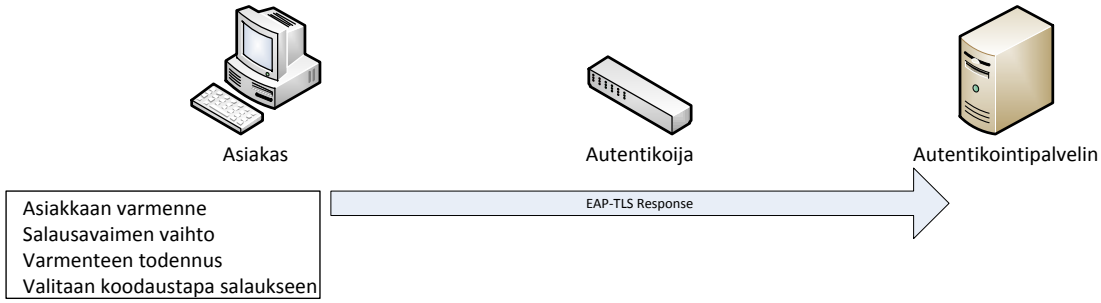
teen voimassaoloajat ja varmenteen julkinen avain sekä sen luomisessa käytetty algoritmi. (Boeyen ym. 2008, 24.)



Kuvio 23. Palvelimen lähettämä varmenne

Mikäli asiakkaan istunnon tunniste on arvoltaan nolla tai se on palvelimelle tuntematon, täytyy palvelimen aloittaa standardin mukaisesti uusi yhteyden muodostaminen. Lisäksi palvelin valitsee käytettävän salausmenetelmän asiakkaan tarjoaman suosituksen perusteella. (Adoba ym. 2008, 4.)

Kolmannen vaiheen aikana asiakas lähettää kuviossa 24 palvelimelle myös oman varmenteensa, julkisen avaimen, varmenteen digitaalisen allekirjoituksen ja salausavaimen yhteyttä varten. Lisäksi mukana on ilmoitus palvelimen varmenteen hyväksymisestä sekä käytettävästä salausmenetelmästä. (Adoba ym. 2008, 4.)



Kuvio 24. EAP-TLS kolmas vaihe

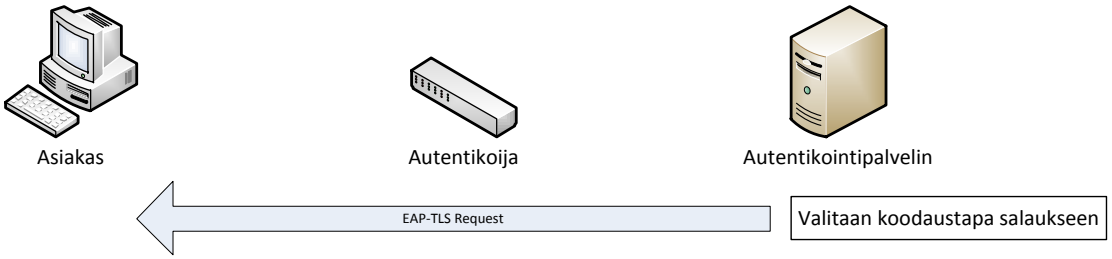
Asiakkaan lähettämä varmenne ja sen myöntäjä näkyvät kuvion 25 Wireshark kuvankaappauksessa. Kuviosta voidaan todeta myös se, että osapuolten varmenteet on myöntänyt sama luotettu taho, joka esiintyi jo aikaisemmassa kuviossa 23, eli tässä tapauksessa luotettu taho on *Oppari CA*.

```
No.  Source            Destination       Protocol  Info
 448 IntelCor_31:68:                      TLSv1   Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
⊞ Frame 448: 1305 bytes on wire (10440 bits), 1305 bytes captured (10440 bits)
⊞ Linux cooked capture
⊟ 802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: EAP Packet (0)
    Length: 1285
  ⊟ Extensible Authentication Protocol
      Code: Response (2)
      Id: 5
      Length: 1285
      Type: TLS EAP (EAP-TLS) (13)
    ⊞ EAP-TLS Flags: 0x80
      EAP-TLS Length: 1275
    ⊟ Secure Sockets Layer
      ⊟ TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
          Content Type: Handshake (22)
          Version: TLS 1.0 (0x0301)
          Length: 1211
        ⊟ Handshake Protocol: Certificate
            Handshake Type: Certificate (11)
            Length: 939
            Certificates Length: 936
          ⊟ Certificates (936 bytes)
              Certificate Length: 933
            ⊟ Certificate (pkcs-9-at-emailAddress=mail@host.domain,id-at-commonName=client,id-at-organizationName=Oppari,id-at-localityName=Jyvasky
              ⊟ signedCertificate
                  version: v3 (2)
                  serialNumber: 2
                ⊟ signature (shaWithRSAEncryption)
                    Algorithm Id: 1.2.840.113549.1.1.5 (shaWithRSAEncryption)
                ⊟ issuer: rdnSequence (0)
                  ⊟ rdnSequence: 6 items (pkcs-9-at-emailAddress=mail@host.domain,id-at-commonName=Oppari CA,id-at-organizationName=Oppari,id-at-lo
                    ⊞ RDNSequence item: 1 item (id-at-countryName=FI)
                    ⊞ RDNSequence item: 1 item (id-at-stateOrProvinceName=Keski-Suomi)
                    ⊞ RDNSequence item: 1 item (id-at-localityName=Jyvaskyla)
                    ⊞ RDNSequence item: 1 item (id-at-organizationName=Oppari)
                    ⊞ RDNSequence item: 1 item (id-at-commonName=Oppari CA)
                    ⊞ RDNSequence item: 1 item (pkcs-9-at-emailAddress=mail@host.domain)
                ⊟ validity
                  ⊞ notBefore: utcTime (0)
                  ⊞ notAfter: utcTime (0)
                ⊞ subject: rdnSequence (0)
                ⊞ subjectPublicKeyInfo
                ⊞ extensions: 6 items
              ⊞ algorithmIdentifier (shaWithRSAEncryption)
                Padding: 0
                encrypted: 50506d3ca6a2c27e3b2ce57b460267399b88e46b6f6f07f2...
```

Kuvio 25. EAP-TLS Response

Ennen viimeiseen vaiheeseen siirtymistä, lähettää palvelin vielä asiakkaalle kuviossa 26 vahvistuksen käytettävästä salausmenetelmästä sekä ilmoituksen siitä, että suo-

jattu TLS-yhteys on muodostettu. (Adoba ym. 2008, 4-5.)



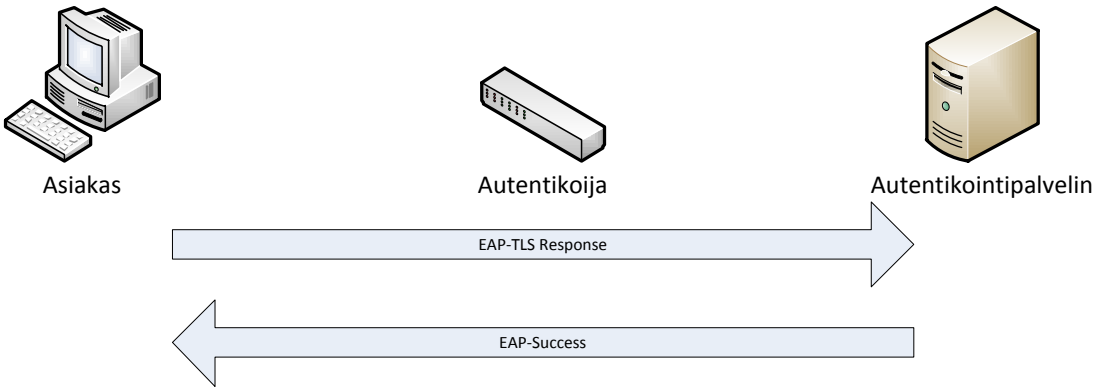
Kuvio 26. EAP-TLS neljäs vaihe

Kyseinen viesti on todennettu alla olevassa kuviossa 27 ja siitä ilmenee palvelimen lähettämä vahvistus käytettävästä salausmenetelmästä.

```
No.  Source          Destination     Protocol  Info
 452 172.16.1.112    172.16.1.1      RADIUS Access-Challenge(11) (id=51, l=127)

⊞ Frame 452: 171 bytes on wire (1368 bits), 171 bytes captured (1368 bits)
⊞ Linux cooked capture
⊞ Internet Protocol Version 4, Src: 172.16.1.112 (172.16.1.112), Dst: 172.16.1.1 (172.16.1.1)
⊞ User Datagram Protocol, Src Port: radius (1812), Dst Port: 56721 (56721)
⊟ Radius Protocol
    Code: Access-Challenge (11)
    Packet identifier: 0x33 (51)
    Length: 127
    Authenticator: 31f4b983f9a8b5ba83e5f3de6400af68
    [This is a response to a request in frame 450]
    [Time from request: 0.000942000 seconds]
  ⊟ Attribute Value Pairs
    ⊟ AVP: l=71  t=EAP-Message(79) Last Segment[1]
        EAP fragment
      ⊟ Extensible Authentication Protocol
          Code: Request (1)
          Id: 6
          Length: 69
          Type: TLS EAP (EAP-TLS) (13)
        ⊟ EAP-TLS Flags: 0x80
            1... .... = Length Included: True
            .0.. .... = More Fragments: False
            ..0. .... = Start: False
          EAP-TLS Length: 59
        ⊟ Secure Sockets Layer
          ⊟ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
              Content Type: Change Cipher Spec (20)
              Version: TLS 1.0 (0x0301)
              Length: 1
              Change Cipher Spec Message
          ⊟ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
              Content Type: Handshake (22)
              Version: TLS 1.0 (0x0301)
              Length: 48
              Handshake Protocol: Encrypted Handshake Message
```

Kuvio 27. EAP-TLS suojattu yhteys

Yhteyden muodostamisen jälkeen siirrytään kuvion 28 mukaisesti todennuksen viimeiseen vaiheeseen.



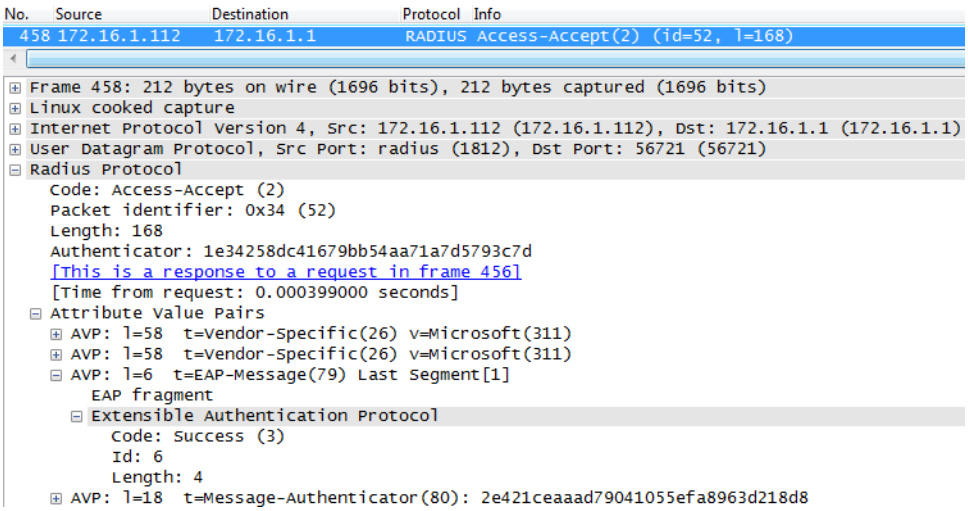
Kuvio 28. EAP-TLS viimeinen vaihe

Asiakas lähettää tässä vaiheessa vahvistuksen palvelimelta saamaansa ilmoitukseen TLS-yhteyden muodostuksesta. Tämä vahvistukseen tarkoitettu viesti on kuviossa 29 ja kyseinen viesti on muotoa EAP-Response. (Adoba ym. 2008, 5-6.)

```
  EAP fragment
⊟ Extensible Authentication Protocol
    Code: Response (2)
    Id: 6
    Length: 6
    Type: TLS EAP (EAP-TLS) (13)
  ⊟ EAP-TLS Flags: 0x00
      0... .... = Length Included: False
      .0.. .... = More Fragments: False
      ..0. .... = Start: False
⊟ AVP: l=18  t=State(24): 4e3e55214a385889ad464be83cc4c33b
  State: 4e3e55214a385889ad464be83cc4c33b
⊟ AVP: l=18  t=Message-Authenticator(80): 1d257a968ed20be4bb027bae2f692f49
  Message-Authenticator: 1d257a968ed20be4bb027bae2f692f49
```

Kuvio 29. EAP-TLS asiakkaan tiedot

Tämän jälkeen palvelin voi hyväksyä asiakkaan pyynnön verkkoon liittymiseksi, lähettämällä kuviossa 30 viestin EAP-Success. Osapuolten välissä oleva verkkolaite näkee tämän vahvistuksen ja avaa portin liikenteelle.



Kuvio 30. EAP-TLS Success

3.3.4 EAP-TTLS

EAP-TTLS (Tunneled Transport Layer Security) on käytännössä laajennus kappaleessa 3.3.3 esiteltyyn EAP-TLS-protokollaan. Vaikka yleisimmät todennusprotokollat ovat suojaustasoltaan hieman heikkoja, mahdollistaa EAP-TTLS kuitenkin päätelaitteen ja käyttäjien todentamisen turvallisesti, koska se suojaa em. protokollat kaappaukselta ja salakuuntelulta. Tämä suojaaminen tapahtuu yksinkertaisesti tunneloimalla todennuksessa käytettävä TLS-istunto. (Funk & Blake-Wilson 2008, 3.)

Asiakas ja todennuksesta huolehtiva palvelin muodostavat TLS-yhteyden, aivan kuten aikaisemmin käsitelty EAP-TLS. Yhteyden muodostamisen jälkeen todennuksessa käytettävät tunnistetiedot salataan, muutetaan attribuuttipareiksi ja siirretään TLS-protokollan Record-kerroksella. (Funk & Blake-Wilson 2008, 13.)

3.3.5 PEAPv2

PEAP (Protected Extensible Authentication Protocol) tarkoittaa käytännössä suojattua EAP-protokollaa ja sen toiminta on lähes samankaltainen EAP-TTLS:n kanssa. Tämä suojaaminen tapahtuu luomalla ensin TLS-yhteys asiakkaan ja autentikointipalvelimen välille. Tämän jälkeen voidaan itse todennus suorittaa turvallisesti käyttämällä jotain muuta EAP-metodia tai suojaukseltaan heikompaa todennusmenetelmää. (Protected EAP Protocol. (Josefsson ym. 2004, 11.)

Kyseinen protokollan aikaisempia versioita ovat PEAPv0 ja PEAPv1. PEAPv2 tuo mukanaan useita parannuksia aikaisempiin protokolliin. Todennusprosessin turvallisuutta on parannettu mm. suojaamalla osapuolten identiteetti, neuvottelumenetelmä sekä neuvottelussa käytettävät viestit. Todennuksen aikana on mahdollista kuljettaa enemmän tietoa, sillä vaihdettava tieto voidaan pakata useampaan kehykseen ja vastaavasti kasata uudelleen vastaanottajan päässä. Lisäksi protokolla optimoi ja nopeuttaa uudelleentodennusta, joka on kaivattu ominaisuus langattomissa lähiverkoissa. (Josefsson ym. 2004, 1-5.)

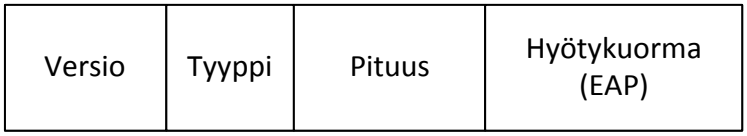
PEAP on alusta alkaen ollut vahvasti tuettuna Windows-käyttöjärjestelmissä, sillä se mahdollistaa suojauksen heikompien protokollien käyttämiselle, jotka puolestaan soveltuvat hyvin käyttäjän tunnistetietojen kyselyyn ja kuljettamiseen. Ensimmäisen kerran PEAP oli tuettuna Windows XP SP1:ssä. (Kamath ym. 2002, 1; PEAP Overview 2012.)

3.4 EAPOL

EAPOL (EAP Over LAN) on toiminnaltaan hyvin yksinkertainen tiedonsiirtoprotokolla ja sen tehtävä onkin nimensä mukaisesti kuljettaa EAP-protokollan sisältämää dataa lähiverkossa. Tällä hetkellä näitä ns. EAPOL-kehyksiä on määritelty yhdeksän kappaletta ja jokaiselle niistä on oma tarkoituksensa. EAPOL-Start, EAPOL-Logoff ja EAPOL-EAP ovat yleisimmin tietoverkossa havaittavia versioita. Muita versioita hyödynnetään lähinnä salausavainten vaihdossa sekä verkkoympäristössä tapahtuviin hälytyksiin. (Port-Based Network Access Control 2010, 90.)

EAPOL-kehys on rakenteeltaan hyvin selkeä ja se sisältää neljä kenttää kuvion 31 mukaisesti. Ensimmäisenä kenttänä on versiokenttä ja sen pituus on yksi tavu. Seuraavana on vuorossa kahden tavun mittainen tyyppikenttä, joka määrittää kehyksen käyttötarkoituksen. Kahden tavun mittainen pituuskenttä ilmoittaa siirrettävän hyötykuorman suuruuden ja mikäli tämän kentän arvo on nolla, kuten kuviossa 32, kehyksessä ei ole mukana hyötydataa. Viimeisenä kenttänä on kehyksessä kuljetettava hyötykuorma, jonka pituus vaihtelee nollasta aina kuljetettavan EAP-kehyksen pituu-

teen. Huomion arvoinen asia on se, että kyseinen kehys ei sisällä tarkistussummaa. (Port-Based Network Access Control 2010, 90-91.)



Kuvio 31. EAPOL-kehys

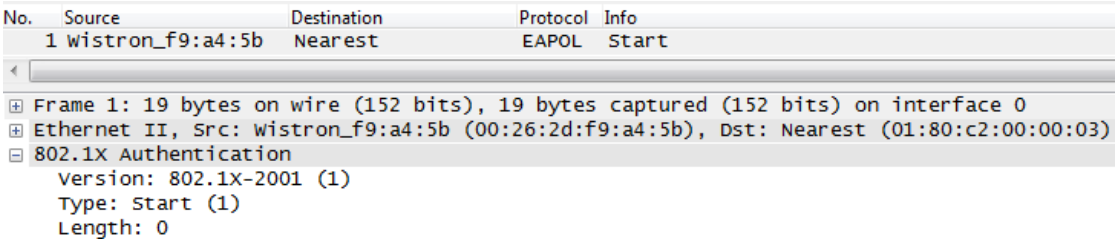
Tärkein kenttä koko EAPOL-kehyksessä on tyyppikenttä, koska se määrittää kehyksen käyttötarkoituksen. Näitä tyyppejä on yhdeksän kappaletta ja ne on lueteltu taulukossa 2**.** Autentikoijan sekä asiakkaan väliset EAP-viestit kuten EAP-Request Identity ovat EAPOL-protokollan sisällä kuljetettavaa hyötydataa ja ne ovatkin tuolloin tyyppiä 0.

Taulukko 2. EAPOL-kehysten tyypit

EAPOL-kehys	**Tyyppi**
EAPOL-EAP	0
EAPOL-Start	1
EAPOL-Logoff	2
EAPOL-Key	3
EAPOL-Encapsulated-ASF-Alert	4
EAPOL-MKA	5
EAPOL-Announcement (Generic)	6
EAPOL-Announcement (Specific)	7
EAPOL-Announcement-Req	8

Tyyppi 1:n kehystä käytetään aloittamaan todennusprosessi asiakkaan toimesta. Tämä tehdään silloin jos asiakas ei jostain syystä saanut autentikoijan lähettämää EAP-

Request Identity -viestiä tai asiakkaan päätelaite on määritelty aloittamaan todennus myöhempänä ajankohtana. Tyyppi 2:n kehystä käytetään päättämään todennusprosessi, eli tarkoittaa käytännössä tilannetta, jossa asiakas lähettää verkkolaitteelle tiedon poistumisestaan tietoverkosta. Tuolloin asiakkaan verkkorajapinta voidaan sulkea liikenteeltä.



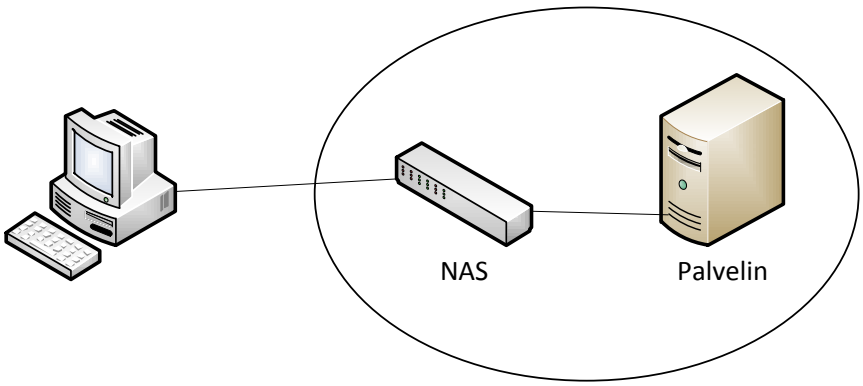
Kuvio 32. EAPOL-Start

4 RADIUS (Remote Authentication Dial In User Service)

4.1 Toiminta

RADIUS on laajasti käytetty tiedonsiirtoprotokolla verkkopohjaisessa käyttäjien sekä laitteiden todennuksessa. Se on esitelty ensimmäisen kerran vuonna 2000 RFC-dokumentissa 2865. Protokollassa on käytännössä kaksi erilaista toimintoa, jotka ovat protokollan perustoiminnot sekä erityisesti kirjanpitoon (accounting) tarkoitettu toiminto. (Rigney ym. 2000, 1; Rigney Livinstong 2000, 8.)

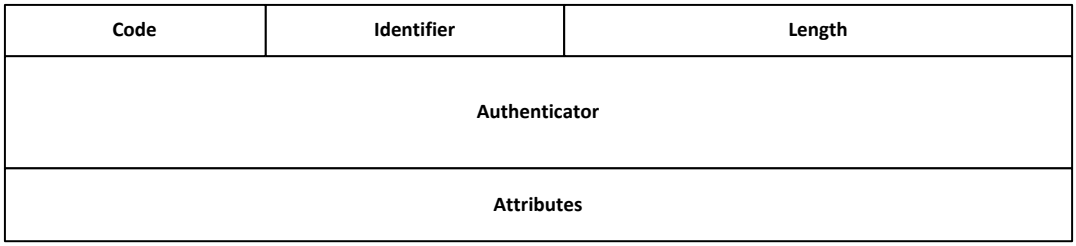
RADIUS toimii ns. palvelin-asiakas-mallin mukaisesti, eikä todennettavaa käyttäjää tai päätelaitetta huomioida tässä palvelumallissa. NAS (Network Access Server) on nimitys asiakaslaitteesta, joka on hyvin usein lähiverkossa käytetty kytkin, kuten kuviossa 33. Asiakkaan tehtävänä on välittää palvelimelle saamansa käyttäjän tai päätelaitteen tunnistetiedot ja palvelimelta saamansa vastauksen perusteella tehdä päätös verkkoresurssien jakamisesta. Palvelimen tehtävänä on ottaa vastaan asiakkaan palvelupyynnöt ja saamiensa tietojen perusteella tehdä vertailu omaan tietokantaansa. Tietokanta koostuu päätelaitteiden tai käyttäjien tunnistetiedoista sekä niille määritellyistä asetuksista. (Rigney ym. 2000, 3)



Kuvio 33. Asiakas-palvelin-malli

4.2 Paketin rakenne

Kuvio 34 esittää RADIUS-pakettia, joka koostuu kentistä Code, Identifier, Lenght, Authenticator ja Attributes. Ensimmäisenä paketissa on Code-kenttä, joka on yhden tavun mittainen ja se määrittää paketin käyttötarkoituksen. (Rigney ym. 2000, 13.)



Kuvio 34. RADIUS-paketti

Code-kentän arvot sekä niitä vastaava RADIUS-paketin käyttötarkoitus on esitelty taulukossa 3**.**

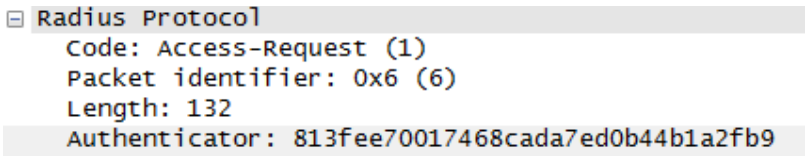
Taulukko 3. Code-kentän arvot ja käyttötarkoitus

Arvo	**Käyttötarkoitus**
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
12	Status-Server (kokeellinen)
13	Status-Client (kokeellinen)
255	Reserved

Nykypäivän tietoverkossa RADIUS-palvelin saattaa saada useita palvelupyyntöjä samanaikaisesti. Jotta palvelin pysyisi tilanteen tasalla vastatessaan oikealle asiakkaalle, on RADIUS-paketissa kahden tavun mittainen Identifier-kenttä, kuten kuvion 35 Wireshark kuvankaappaus havainnollistaa. Kentän tarkoituksena on yksilöidä jokainen palvelutapahtuma. Lisäksi palvelin kykenee tunnistamaan päällekkäisiä palvelupyyntöjä lyhyen ajanhetken sisällä perustuen lähettäjän IP-osoitteeseen, lähdeporttiin sekä RADIUS-paketin tunnistetietoon. (Rigney ym. 2000, 13.)

Lenght-kentän tehtävä on yksinkertaisesti määrittää kuinka suuri kyseinen RADIUS-paketti on kokonaisuudessaan, mukaan luettuna kaikki paketissa olevat kentät. Itse kentän koko on kaksi tavua ja mikäli sen pituus on suurempi, tulee kyseinen paketti käsitellä kuin se olisi loppuosaltaan vain täytettä. Mikäli paketin koko sattuu olemaan lyhyempi kuin lenght-kenttä kertoo, paketti hylätään. (Rigney ym. 2000, 14.)

Authenticator-kenttä on pituudeltaan 16 tavua ja kaikista merkitsevin tavu siirretään ensimmäisenä. Tätä kenttää käytetään todentamaan RADIUS-palvelimen vastaus ja sitä myös käytetään salasanan piilottamiseen. Vaikka kenttä onkin paketissa yhtenäinen, voidaan sen toiminta ajatella koostuvan kahdesta osasta, pyynnöstä sekä vastauksesta. Mikäli kytkimen on tarkoitus pyytää todennusta, eli sen lähettämä RADIUS-paketti on kuvion 35 mukaisesti tyyppiä Access-request, generoidaan Authenticator-kentän sisältö sattumanvaraiseksi arvoksi. Tämän arvon pitäisi olla uniikki ja ennustamaton koko todennusprosessin ajan. (Rigney ym. 2000, 14.)



Kuvio 35. RADIUS Access-Request

Vastatakseen asiakkaan pyyntöihin on RADIUS-paketin tyyppi jokin seuraavista: Access-Accept, Access-Reject tai Access-Challenge. Tuolloin Authenticator-kentän sisältämä arvo on MD5-algortimilla muodostettu tiiviste koko paketin sisältämistä kentistä, attribuuteista sekä tunnussanasta. (Rigney ym. 2000, 15.)

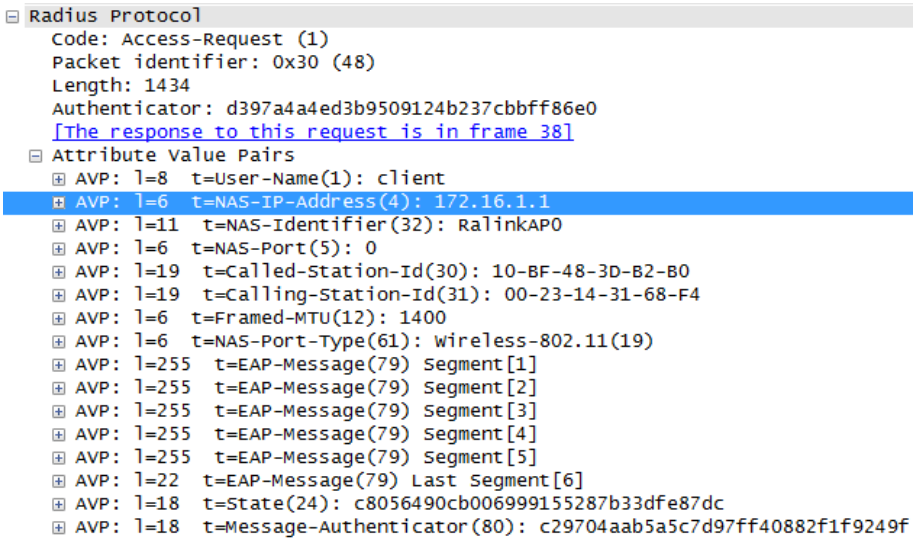
Attributes-kenttä on pituudeltaan joustava ja se saattaa sisältää huomattavan määrän erilaisia attribuutteja. RADIUS-paketti voi sisältää myös ylimääräisiä attribuutteja, mutta palvelimen ei ole mikään pakko ottaa niitä huomioon. (Rigney ym. 2000, 21-22.)

4.3 Tyypit

4.3.1 Access-Request

Mikäli päätelaite tai käyttäjä haluaa liittyä tietoverkkoon, välittää kytkin palvelimelle RADIUS-paketin, jonka Code-kentän arvo on 1. Kyseessä on siis tuolloin Access-Request -viesti, joka pitää sisällään kuvion 36 mukaisesti User-Name, NAS-IP-Address ja NAS-Identifier -attribuutit. Mikäli käyttäjän salasana esiintyy paketissa, pitää kyseisessä paketissa olla myös User-Password tai CHAP-Password -attribuutti, mutta ei molempia samaan aikaan. Siirrettävät salasanat suojataan se MD5-algoritmilla.

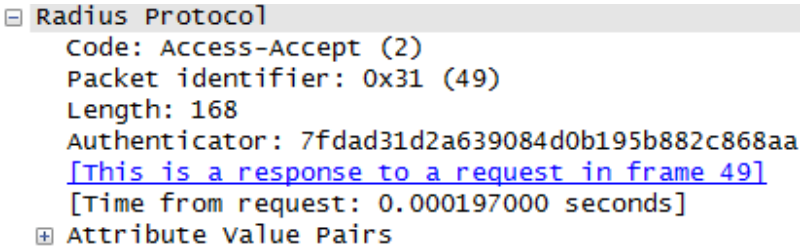
Jokainen kerta kun Access-Request -viestiin vastataan palvelimen toimesta, muuttuvat myös Identifier ja Attributes -kenttien arvot. Kuitenkin uudelleen lähetyksessä Identifier-kentän arvon tulee pysyä muuttumattomana. (Rigney ym. 2000, 16-17.)



Kuvio 36. RADIUS Access-Request

4.3.2 Access-Accept

Palvelimen saadessa asiakkaalta Access-Request -paketin, jossa kaikki attribuutit täsmäävät sen omaan tietokantaan. Palvelimen täytyy hyväksyä asiakkaan pyyntö lähettämällä viesti Access-Accept Code-kentän arvolla 2. Kyseinen paketti on kaapattu kuviossa 37 palvelimen ja asiakkaan väliltä. Kaappauksessa käytettiin ohjelmaa Wireshark**.** Paketin Identifier-kentän arvo on saman kuin asiakkaan lähettämässä Access-Reques -viestissä, koska juuri tähän pyyntöön ollaan vastaamassa myönteisesti. (Rigney ym. 2000, 17-18.)



Kuvio 37. RADIUS Access-Accept

4.3.3 Access-Reject

Jos yksikin asiakkaan lähettämistä attribuuteista ei täsmää palvelimen tietokannassa oleviin tietoihin, pitää palvelimen hylätä asiakkaan pyyntö. Tämä tapahtuu lähettämällä paketti Access-Reject, Code-kentän arvolla 3. Tämä asiakkaalle lähetetty viesti voi sisältää useamman attribuutin, jotka asiakas välittää päätelaitteelle tai käyttäjälle. (Rigney ym. 2000, 19.)

4.3.4 Accounting-Request

RFC 2866 määrittelee RADIUS-protokollan toiminnot tapahtumien valvontaan sekä kirjanpitoon. Accouting-Request sekä Accouting-Response on tarkoitettu ilmoittamaan ja kirjaamaan tietoverkon tapahtumia. Käyttäjän kirjautuminen verkkolaitteen hallintaan, muutosten teko järjestelmään, uudelleen käynnistykset yms. ovat lokitapahtumia, jotka olisi hyvä kerätä talteen keskitetysti. RADIUS-palvelin tukee kyseistä toimintoa ja tapahtumista ilmoittamiseen on olemassa oma viestityyppinsä.

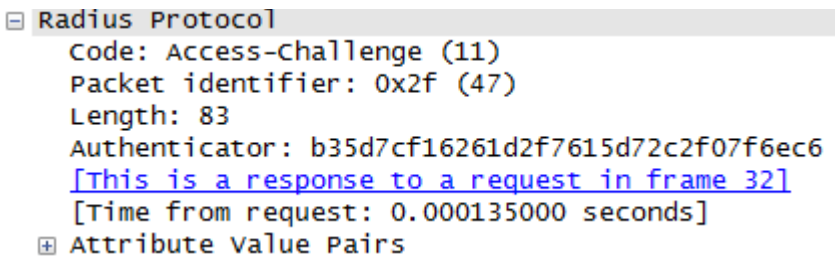
Accouting-Request -paketti merkataan Code-kentän arvolla 4 ja pakollisia attribuutteja on NAS-IP-Address tai NAS-Identifier. Näiden lisäksi viestin tulisi sisältää attribuutit NAS-Port tai NAS-Port-Type. Mikäli RADIUS-palvelin suorittaa tapahtuman kirjaamisen onnistuneesti, lähetetään siitä vastaus. Standardin mukaisesti RADIUS-palvelin ei saa lähettää mitään vastausta, mikäli kirjaaminen epäonnistuu. (Rigney Livinstong 2000, 7.)

4.3.5 Accounting-Response

Onnistuneesta lokikirjauksesta RADIUS-palvelin lähettää vastauksen viestillä Accouting-Response ja kyseisen paketin Code-kentän arvo 5. Vastauksen Identifier-kentän pitää kertoa, mikä tapahtuma saatiin onnistuneesti kirjattua. (Rigney Livinstong 2000, 8.)

4.3.6 Access-Challenge

Tilanteessa jossa RADIUS-palvelin haluaa lisätietoa käyttäjästä tai päätelaitteesta, tulee palvelimen lähettää asiakkaalle haaste paketilla Access-Challenge ja Code-kentän arvolla 11, kuten kuviossa 38 on nähtävissä. Asiakas välittää tämän haasteen käyttäjälle ja vastauksen saatuaan lähettää sen takaisin palvelimelle Access-Response -pakettina. Kyseinen haaste voi sisältää useamman attribuutin, mutta vain tietyt standardissa määritetyt attribuutit ovat sallittuja. (Rigney ym. 2000, 21.)



Kuvio 38. RADIUS Access-Challenge

5 Työympäristö

5.1 Työympäristö

Yrityksen palvelimet ja työasemat oli varustettu Windows 7 sekä Windows 2008 R2 -käyttöjärjestelmillä. Tästä syystä niiden hallinnointi on suhteellisen vaivatonta pienellä kin työvoimalla. Työn määrään vaikuttaa tietysti toimialueen rakenne, mitä palveluita siihen on sisällytettynä ja kuinka hyvin tunnetaan käytettävät työkalut.

Windows Active Directory sekä muut toimialueen palvelut ovat yhdessä kuin suuri palapeli ja sen tehokkaaseen hallintaan sekä oppimiseen saattaa mennä hyvinkin kauan aikaa. Tästä syystä käsitellään seuraavaksi vain pintapuolisesti tähän opinnäytetyöhön liittyviä palveluita.

5.1.1 Active Directory Domain Services

Active Directory Domain Services (AD DS) on Microsoftin kehittämä hallintatyökalu ja sen tarkoituksena on yksinkertaisesti helpottaa toimialueeseen liitettyjen resurssien, kuten esimerkiksi käyttäjien sekä työasemien hallintaa. Windows 2008 R2:ssa Active Directory (AD) eli aktiivihakemisto on erikseen asennettava rooli ja se tuo mukanaan useita hallintatyökaluja sekä palveluita. (Minasi 2010, 227-229.)

Aktiivihakemisto on toimialueen runkopalvelu ja lähes kaikki, myös tässä opinnäytetyössä käytetyt palvelut, hyödyntävät sitä tavalla tai toisella. Aktiivihakemisto pitää nimensä mukaisesti kirjaa kaikista siihen liitetyistä objekteista. Objekti on tässä tapauksessa verrattavissa todellisiin käyttäjiin sekä heidän käyttämiinsä työasemiin. Aktiivihakemiston tehtävänä on yksinkertaistaa sekä helpottaa näiden objektien hallintaa. Jokaiselle käyttäjälle luodaan oma käyttäjätunnus ja se sidotaan käyttäjäryhmiin. Näillä käyttäjäryhmillä yksinkertaistetaan käyttöoikeuksien hallintaa sekä resurssien jakamista, tällaisia tapauksia ovat esimerkiksi oikeus käyttää verkkotulostinta ja pääsy vain työtehtävän vaatimiin tiedostoihin. (Minasi 2010, 227-229.)

5.1.2 Active Directoryn looginen rakenne

Active Directoryn toimialue sijaitsee niin sanotusti metsässä (forest) ja se on joko yhden tai useamman toimialueen muodostama ryhmä. Metsän juuritoimialue on aina ensimmäisenä asennettu toimialue ja tämä on kaikista tärkein toimialue koko toimialuemetsässä. (Minasi 2010, 230.)

Toimialue on kokoelma työasemia, jotka jakavat yhteisen tietokannan. Ylimmän toimialuenimen määrittää Active Directoryn nimiavaruus. Toimialueen nimistä on huomioitava se seikka, että tismalleen samaa nimeä ei voi käyttää metsän sisällä. (Minasi 2010, 230.)

Toimialuemetsä sisältää ns. yleisen luettelon, joka sisältää kaikki metsään kuuluvat objektit. Tämän luettelon ansiosta toimialuemetsän sisällä on helppo etsiä mitä tahansa tietoa. (Minasi 2010, 230.)

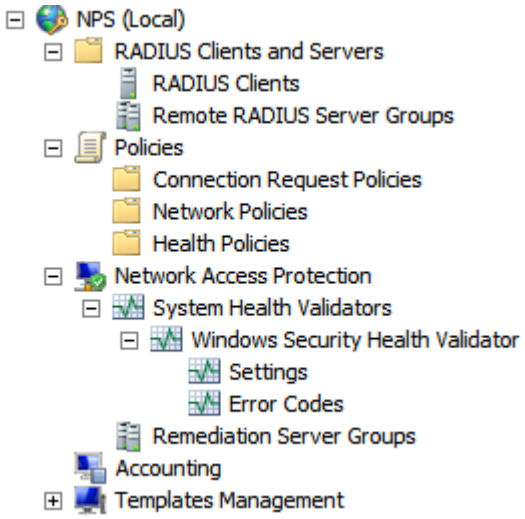
Metsä sisältää puita (tree), jotka ovat kokoelma toimialueita saman DNS -nimiavaruuden alla. Nämä toimialueet muodostavat hierarkkisen rakenteen. Ylin toimialue on niin sanottu juuritoimialue (root domain) ja sen alla olevia toimialueita kutsutaan alitoimialueiksi (child domain). (Minasi 2010, 230.)

Toimialueiden sisälle voidaan muodostaa vielä aliryhmiä, jotka helpottavat ympäristön hallintaa. Näitä aliryhmiä kutsutaan organisaatioyksiköiksi (OU) ja ne on tehty ainoastaan hallinnallisista syistä. Organisaatioyksiköiden käyttäminen ei ole pakollista, mutta niiden avulla voidaan linkittää ryhmäkäytännöt (GP) koskemaan esimerkiksi tiettyä henkilöstöä tai toimipistettä. (Minasi 2010, 230.)

5.1.3 Network Policy and Access Services

Network Policy and Access Services (NPAS) -rooli mahdollistaa tietoverkon turvaamisen. Kyseinen rooli sisältää seuraavat työkalut: Network Policy Server (NPS), Health Registration Authority (HRA) sekä Host Credential Authorization Protocol (HCAP). (Network Policy and Access Services 2014.)

NPS on käytännössä RADIUS-palvelin Windows 2008 R2:ssa, kuten kuvio 39 osoittaa. NPS mahdollistaa mm. työasemien ja käyttäjien tunnistamisen sekä käyttöoikeuksien määrittämisen niitä pyydettäessä. Hyvänä esimerkkinä tästä on opinnäytetyössä tehty työasemien todennus sekä niiden päästäminen yrityksen tietoverkkoon, mikäli vaadittavat ehdot täyttyvät. (Davies & Northrup 2008, 578.)



Kuvio 39. Network Policy Server

5.1.4 Windows Server Certificate Services

Windows Server 2008 R2 tarjoaa Active Directory Certificate Services -roolin, joka mahdollistaa varmenteiden hallinnan toimialueen sisällä sekä oman julkisen avaimen infrastruktuurin pystyttämisen. Kyseinen rooli on vahvasti sisällytetty toimialueen palveluiden kanssa ja lisäksi sen hallinnointi on tehty mahdollisimman helpoksi, joten sen käyttäminen on vähintäänkin suositeltavaa. (Minasi 2010, 133.)

Kyseinen rooli voidaan asentaa kahdella tavalla, joko Enterprise CA:ksi tai Standalone CA:ksi. Standalone CA ei ole riippuvainen aktiivihakemistosta ja se toimii offline-tilassa, lisäksi siinä on vähemmän automatisoituja toimintoja. Enterprise CA on taas riippuvainen aktiivihakemistosta, mutta samalla se mahdollistaa varmenteiden jakamisen toimialueen työasemille automaattisesti. (Minasi 2010, 239.)

5.1.5 Group Policy

Ryhmäkäytäntö (GP) on yksi toimialueen hallintaan tarkoitetuista työkaluista. Se mahdollistaa tietyn asetuksen määrittämisen vain kerran ja samalla se kopioidaan useammalle käyttäjälle tai työasemalle.

Esimerkiksi tilanteessa, jossa palomuuri pakotetaan päälle työasemissa, ainoat toimenpiteet ovat ryhmäkäytäntöobjektin (GPO) luominen ja kyseisen objektin liittäminen haluttuun organisaatioyksikköön (OU). Tämän jälkeen kyseinen asetus menee kaikkiin organisaatioyksikössä oleviin työasemiin.

Ryhmäkäytäntöobjektit voidaan liittää koskemaan organisaatioyksikköä sekä toimialuetta. Kun toimialue perustetaan, luodaan samalla kaksi oletusryhmäkäytäntöobjektia, jotka ovat default domain policy ja default domain controllers policy. (Minasi 2010, 229.)

5.2 Julkisen avaimen infrastruktuuri

5.2.1 Yleistä

Julkisen avaimen infrastruktuuri (PKI) on kokoelma organisointimenetelmiä, protokollia sekä prosesseja. Näiden tehtävänä on huolehtia julkisten avainten luomisesta, varmentamisesta ja jakamisesta. PKI on toiminnaltaan hyvin joustava, sillä se kykenee tarjoamaan julkisia avaimia sekä varmentaman niitä aina yksittäisestä käyttäjästä kokonaiseen valtioon. (Smith 2001, 415.)

PKI mahdollistaa suojatun tiedon siirtämisen sekä menetelmän todentaa mm. käyttäjiä ja päätelaitteita. Tämä tapahtuu käyttämällä julkisesta sekä salaisesta avaimesta koostuvaa avainparia. Tämä kyseinen avainpari linkitetään matemaattisesti toisiinsa ja menetelmää kutsutaankin epäsymmetriseksi salaukseksi. Menetelmän idea perustuu siihen, että esimerkiksi salaisella avaimella koodattu tieto voidaan purkaa vain saman avainparin julkisella avaimella. Tämän salauksen vahvuus perustuu siihen, että matemaattisesti on hyvin hankalaa tai jopa mahdotonta päätellä salatusta tiedosta avainparin toista osapuolta. (Garman 2003, 208.)

Käytettäessä epäsymmetristä avainparia, tulee julkisen avaimen olla kaikkien niiden osapuolten tiedossa joiden kanssa halutaan vaihtaa tietoa. Tämä siitä syystä, että lähettäjän salaisella avaimella salattu tieto, voidaan purkaa vain avainparin julkisella avaimella. Mikäli salainen avain joutuu muiden tietoon, on olemassa vaara, että tietoturvaa uhkaavalla taholla on myös hallussa julkinen avain. Tästä syystä hallussa oleva salainen avain tulisi pitää muiden ulottumattomissa. (Garman 2003, 208.)

PKI mahdollistaa myös sähköisen allekirjoituksen, joka perustuu samaan menetelmään kuin epäsymmetrinen avainpari. Allekirjoituksella voidaan varmistua toisesta osapuolesta. Allekirjoittaakseen lähetettävän tiedon, lähettäjä käyttää omaa salaista avaintaan. Jos vastaanottaja kykenee purkamaan saamansa tiedon lähettäjän julkisella avaimella, voidaan tietoon sekä lähettäjään luottaa. (Garman 2003, 209.)

5.2.2 Varmenne

Julkisen avaimen infrastruktuuri käyttää digitaalisia sertifikaatteja, eli varmenteita laitteiden, käyttäjien sekä palveluiden varmentamisessa. RFC 5280 määrittää varmenteen pakolliset, vaihtoehtoiset ja vapaasti valittavat kentät. Yleisen yhteensopivuuden kannalta on erittäin tärkeää, että varmenteet tehdään standardin mukaisesti. Yleisin ja tuetuin muoto varmenteissa on nykyään X.509. X.509-varmenteesta on olemassa kolme versiota ja suurimmat erot näiden välillä ovat uudet tietuekentät. (Boeyen ym. 2008, 9.)

5.2.3 Certificate Authority

Julkisen avaimen infrastruktuuri luottaa kolmanteen osapuoleen, minkä tehtävänä todentaa ja asettaa todennuksessa olevat osapuolet luotetuiksi. Tästä kolmannesta osapuolesta käytetään nimitystä luotettu taho (CA). CA myöntää varmenteita rekisteröintielimeltä saapuvien pyyntöjen perusteella. Varmenteen myöntäminen tapahtuu siten, CA allekirjoittaa myönnettävän varmenteen digitaalisesti salaisella avaimellaan.

Julkisen avaimen infrastruktuuri rakentuu niin sanottuun luottoketjuun, minkä päässä on aina juurivarmentaja (RootCA). Tästä syystä juurivarmentaja ei voi luottaa kuin itseensä ja niinpä se allekirjoittaa oman varmenteensa. (Choudhury 2002, 30.)

6 Työn toteutus

6.1 Yleistä

Toteutus alkoi tutustumalla käytettävissä oleviin aktiivilaitteisiin, niiden ominaisuuksiin sekä kertaamalla hieman porttikohtaisen todennuksen teoriaa. Seuraavana vuorossa oli tutustuminen autentikointipalvelimen toimintaan ja sen mahdollisuuksiin. Tämä jälkeen vuorossa oli suunnitelman tekoa, kuinka aktiivilaitteet sekä autentikointipalvelimet tulisi toteuttaa verkkoympäristöön. Lähtökohtana oli saada aikaiseksi mahdollisimman käyttäjäystävällinen, vikasietoinen sekä henkilöstölle vähän lisätöitä tuova järjestelmä.

Verkkolaitteiden eli tässä tapauksessa kytkimien valmistelu oli lopulta hyvin yksinkertainen toimenpide, kunhan vain tarvittavat komennot sekä niiden testaaminen saatiin suoritettua. Autentikointipalvelimen testauksessa meni taas huomattavasti enemmän aikaa, mutta kaiken kaikkiaan senkin pystyttäminen oli suhteellisen nopeata, johtuen valmistajan kattavasta materiaalista. Työasemien osalta sopivien asetusten hakeminen sekä testaus suoritettiin ensin yhdellä työasemalla, kunnes toimivat asetuksen löytyivät. Tämän jälkeen tehtiin uusi ryhmäkäytäntö, jonka tehtävänä oli viedä uudet asetukset työasemille. Tämä ryhmäkäytäntö asetettiin vaiheittain koskemaan useampia työasemia.

Suurimmat ongelmat tässä työssä oli saada ryhmäkäytännöt toimimaan porttikohtaisen todennuksen kanssa. Tämä oli todella tärkeää, koska ryhmäkäytännöt helpottavat huomattavasti henkilöstön työmäärää.

Ensimmäisen kytkimen toimiessa moitteettomasti porttikohtaisen todennuksen kanssa, lisättiin todennuksen alle vaiheittain lisää työasemia. Testaaminen jatkui mm. jakamalla asennustiedostoja ryhmäkäytäntöjen avulla sekä testaamalla toissijaisen autentikointipalvelimen toimivuutta mahdollisessa vikatilanteessa. Tämän ns. testausjakson päätyttyä, lisättiin myös muiden paikkakuntien toimipisteet käyttämään porttikohtaista todennusta.

6.2 Aktiivilaitteiden konfigurointi

Työn toteutus alkoi määrittämällä yrityksen tietoverkossa oleviin aktiivilaitteisiin eli kytkimiin RADIUS-palvelimien parametrit, kuten palvelimen IP-osoitteet sekä laitekohtaisesti jaetut tunnussanat. Kyseiset toiminnot suoritettiin komennoilla:

#radius-server host <IP-osoite>

#radius-server key <tunnussana>

Syötetyt asetukset tarkistettiin vielä komennolla *#show radius* ja komennon antama tuloste näkyy kuviossa 40. Tässä työssä päädyttiin lopulta käyttämään globaalia tunnussanaa laitekohtaisesti, vaikka mahdollisuus olisi ollut käyttää myös laite sekä palvelinkohtaista tunnussanaa. Syynä tähän oli helpottaa muutosten kopioimista toissijaiselle palvelimelle. Käytetyn tunnussanan tuli olla sama, kuin autentikointipalvelimella määritetty asiakaskohtainen tunnussana. Mikäli ensisijainen autentikointipalvelin ei jostain syystä vastaa kytkimen pyyntöihin, siirtyy kytkin tuolloin käyttämään toissijaista autentikointipalvelinta. (Command Line Interface Reference Guide. 2007, 282.)

```
nw8  # show radius

 Status and Counters - General RADIUS Information

  Deadtime(min) : 0
  Timeout(secs) : 5
  Retransmit Attempts : 3
  Global Encryption Key :  xxxxxx
  Dynamic Authorization UDP Port : 3799
  Source IP Selection : Outgoing Interface

                  Auth Acct DM/ Time    |
  Server IP Addr  Port Port CoA Window | Encryption Key                    OOBM
  --------------- ---- ---- --- ------ + -------------------------------- ----
  172. xxx.xxx .119  1812 1813 No  300    |                                  No
  172.xxx.xxx .121  1812 1813 No  300    |                                  No
```

Kuvio 40. RADIUS-palvelimien asetukset

Seuraavaksi määritettiin kytkimet ja niiden työasemakohtaiset portit käyttämään porttikohtaista todennusta. Kyseiset toimenpiteet suoriettiin komeinnoilla:

#aaa authentication port-access eap-radius

#aaa port-access authenticator <portti>

#aaa port-access authenticator active

Komento *#show port-access summary* näyttää kuviossa 41 yhteenvedon kytkimen porttikohtaisista asetuksista. Kyseisestä kuviosta näkyy myös se, että porttikohtainen todennus on päällä ja se on aktiivisena portissa A2. Tarvittaessa porttikohtainen todennus saadaan nopeasti pois päältä komennolla *#no aaa port-access authenticator active*. (Command Line Interface Reference Guide. 2007, 24-25.)

```
nw8  # show port-access summary

 Port Access Status Summary

  Port-access authenticator activated [No] : Yes
  Allow RADIUS-assigned dynamic (GVRP) VLANs [No] : No

  Note: * indicates values dynamically overridden by RADIUS.

       |    Authenticator    |   Web Auth    |   MAC Auth
  Port | Enabled Mode  Limit | Enabled Limit | Enabled Limit
  ---- + ------- ----- ----- + ------- ----- + ------- -----
  A1   | No      Port  0     | No      1     | No      1
  A2   | Yes     Port  0     | No      1     | No      1
  A3   | No      Port  0     | No      1     | No      1
```

Kuvio 41. Yhteenveto porttikohtaisen todennuksen asetuksista

Porttikohtaisen todennuksen toimiessa, todennetut työasemat tarkistettiin komennolla *#show port-access authenticator clients***.** Komennon antama syöte näkyy kuviossa 42 ja siitä selviää myös onnistuneesti todennetut työasemat, niiden tila, MAC-osoite sekä minkä portin takaa ne ovat liittyneenä tietoverkkoon. (Command Line Interface Reference Guide. 2007, 63.)

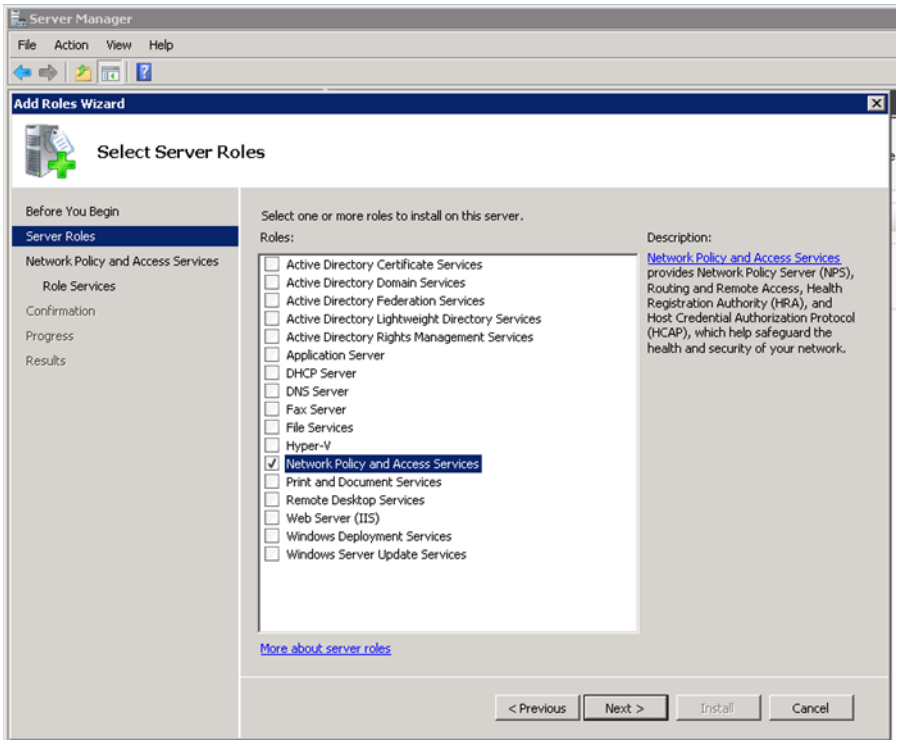
```
nw8  # show port-access authenticator clients

 Port Access Authenticator Client Status

  Port-access authenticator activated [No] : Yes
  Allow RADIUS-assigned dynamic (GVRP) VLANs [No] : No

  Port Client Name           MAC Address   IP Address      Client Status
  ---- --------------------- ------------- --------------- --------------------
  A2   host/WS1005.xxxxxxxx ... 6c3be5-f23a95 n/a           Authenticated
  A10  host/WS808.xxxxxxxx  ... 643150-415f92 n/a           Authenticated
```

Kuvio 42. Todennetut työasemat

6.3 Autentikointipalvelimen asennus

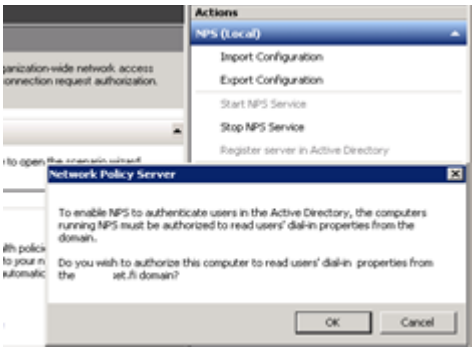
6.3.1 Network Policy and Access Services

Autentikointipalvelimelle asennettiin kuvion 43 mukaisesti Network Policy and Access Services -rooli ja asennus suoritettiin käyttämällä työkalua Server Manager. Tämä rooli tuo mukanaan Network Policy Serverin (NPS) ja sen käyttäminen mahdollistaa mm. työasemien tunnistamisen sekä asetusten määrittämisen.



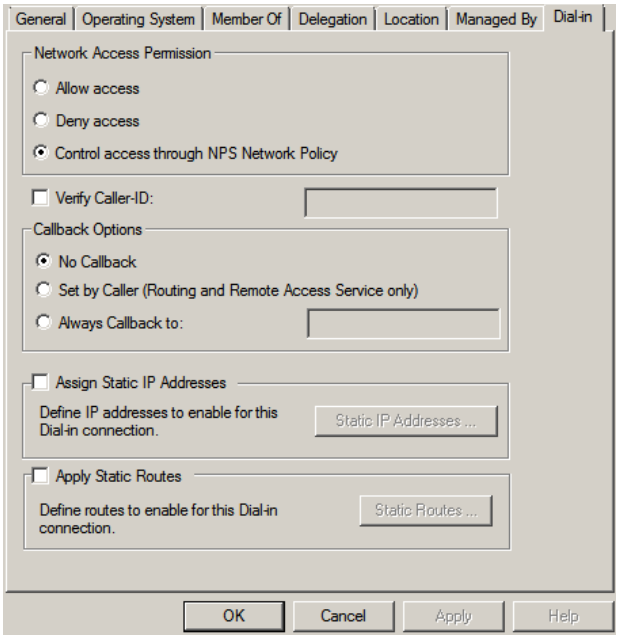
Kuvio 43. Network Policy and Access Services -roolin asennus

Roolin asennuksen jälkeen rekisteröitiin NPS kuvion 44 mukaisesti aktiivihakemiston alle. Tämä tehtiin siitä syystä, että NPS pääsisi tutkimaan aktiivihakemistossa olevien käyttäjien sekä työasemien Dial-in -välilehdellä määritettyä asetusta. Oletuksena kyseinen asetus määrittää pääsyn tietoverkkoon Network Policyn kautta. (Davies & Northrup 2008, 388.)



Kuvio 44. NPS:n rekisteröinti aktiivihakemistoon

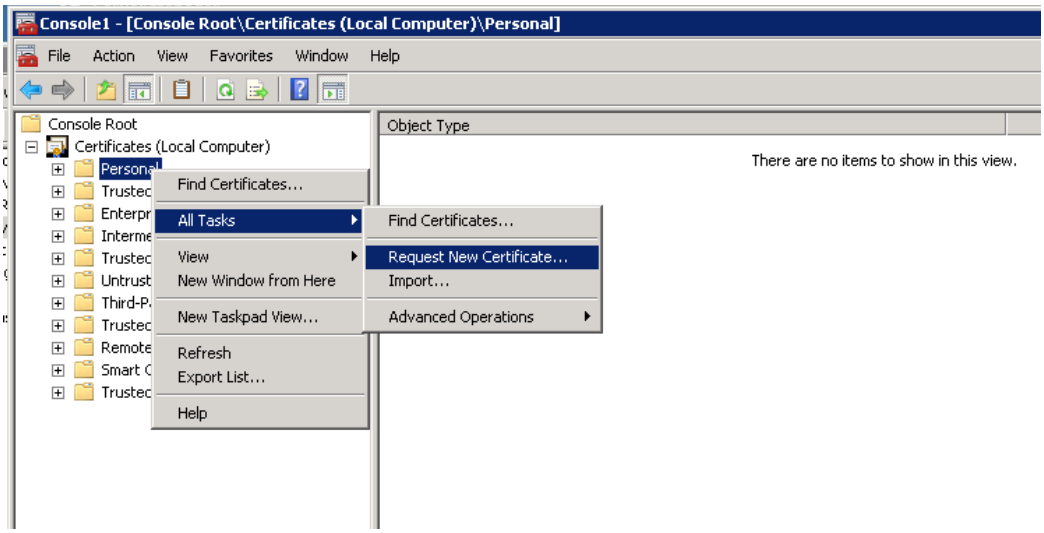
Alla olevassa kuviossa 45 on esitelty aktiivihakemistossa olevan työaseman Dial-in -välilehti, joka oletuksena käyttää NPS-palvelinta verkkoyhteyden hallinnoimiseksi, myös käyttäjiltä löytyy vastaava välilehti ja asetus.



Kuvio 45. Käyttäjien Dial-in -asetukset

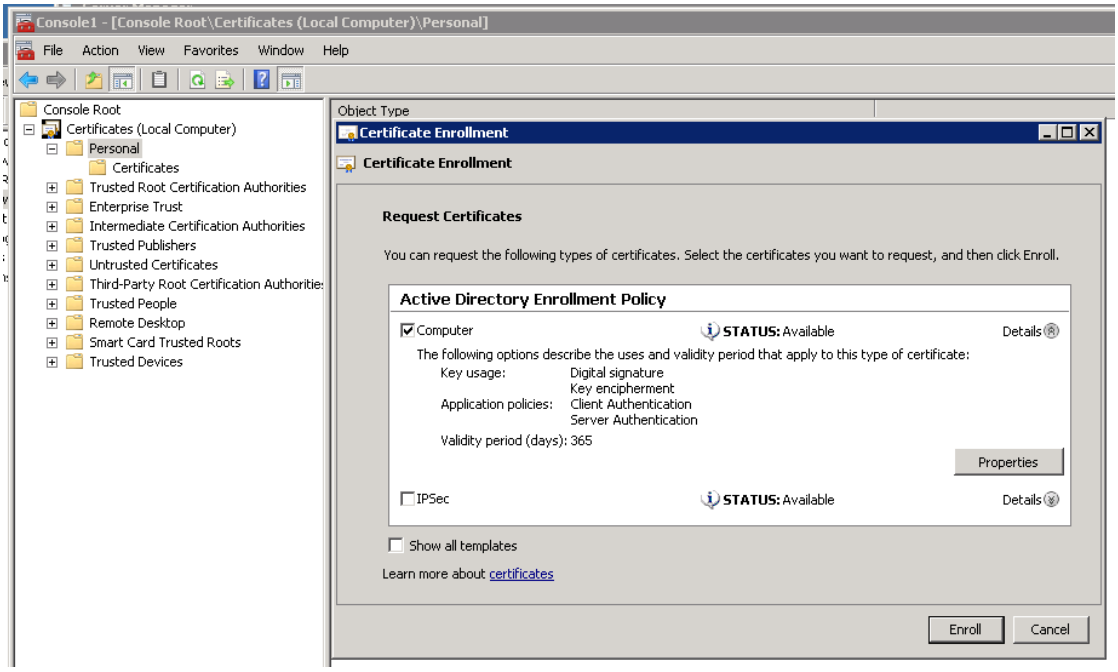
6.3.2 Palvelinkohtainen varmenne

Asennuksen jälkeen haettiin palvelimelle oma varmenne. Tätä varmennetta tultaisiin käyttämään EAP-TLS:n kanssa. Kyseinen toimenpide suoritettiin käyttämällä Microsoft Management Consolea (MMC) kuviossa 46. (Davies & Northrup 2008, 271.)



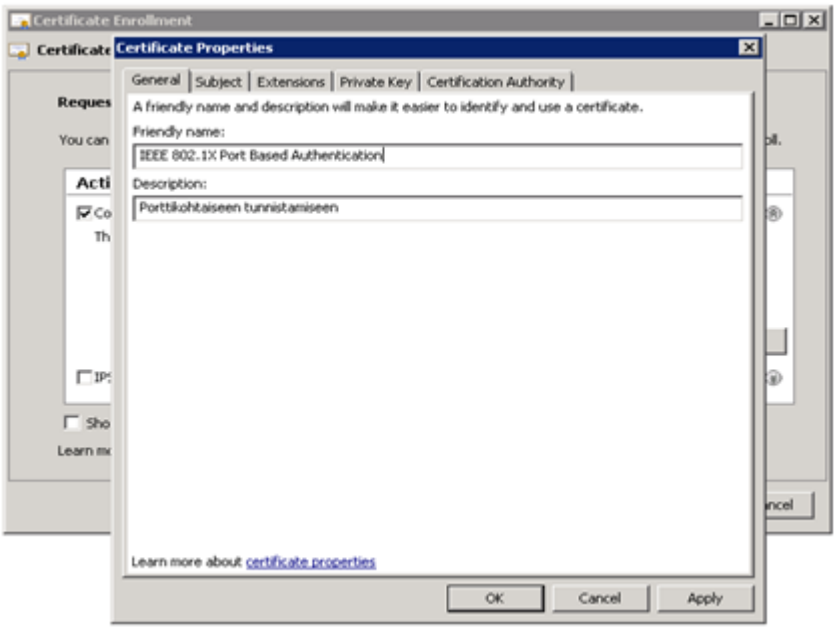
Kuvio 46. Microsoft Management Console

Varmenne haettiin tässä tapauksessa paikalliselle koneelle ja se on kuvion 47 mukaisesti voimassa yhden vuoden sekä on tarkoitettu käytettäväksi todennuksessa.



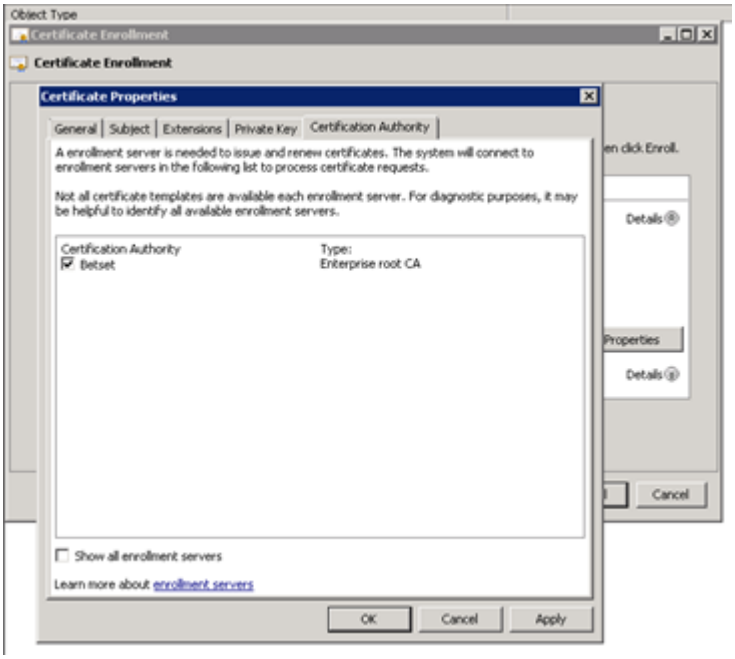
Kuvio 47. Varmenteen voimassaoloaika

Varmenteen asetuksista voidaan lisätä kuvaava nimi, mikä myöhemmin helpottaa kyseessä olevan varmenteen tunnistamista ja käyttöä. Kyseinen ominaisuus löytyy kuvion 48 mukaisesti varmenteen general-välilehdeltä.



Kuvio 48. Varmenteen kuvaus

Ennen varmenteen pyytämistä todettiin vielä kuviossa 49, että varmenteen myöntäjä on yrityksen oma Certificate Authority eli CA.

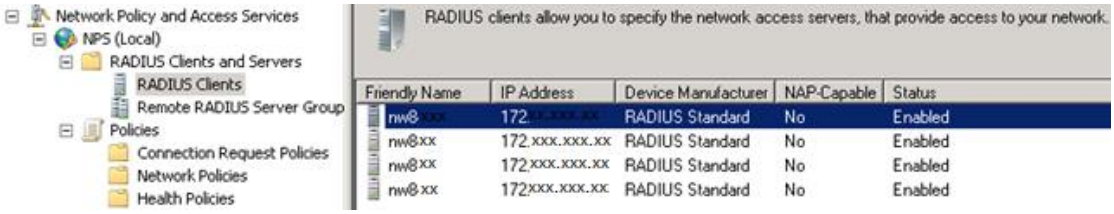


Kuvio 49. Varmenteen myöntäjä

Myös työasemille voitaisiin hakea varmenteet samalla tavalla, mutta ne saavat oman varmenteensa automaattisesti, määritetyn politiikan ja toimialueen kautta. (Minasi 2010, 133.)

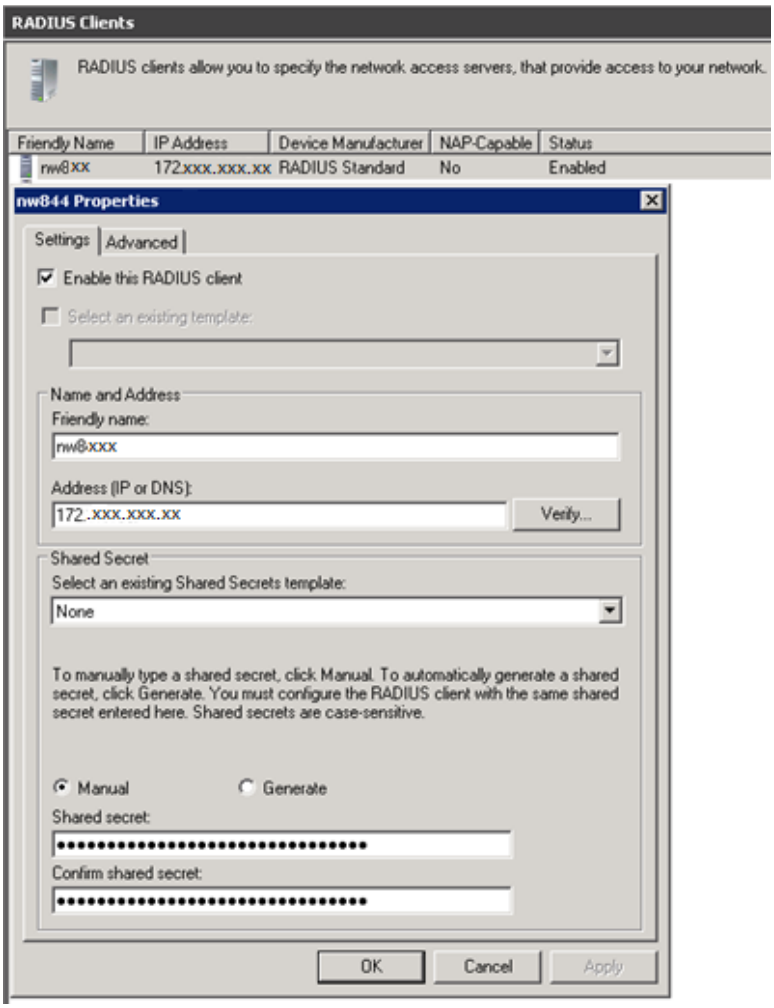
6.3.3 Network Policy Server

Seuraavaksi määritettiin tarvittavat politiikat, minkä mukaan Network Policy Server (NPS) sallii kytkimien pyynnöt ja mitkä vaatimukset työasemien tulee täyttää tietoverkkoon päästäkseen. Lisäksi yrityksen kytkimet lisättiin oman hakemistonsa alle, kuten kuvio 50 havainnollistaa.



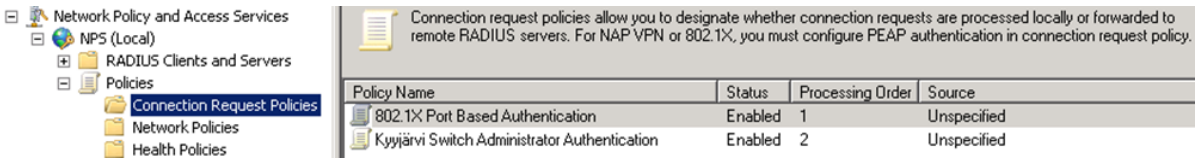
Kuvio 50. RADIUS-asiakkaat

Jokaiselle kytkimelle määritettiin oma tunnussana kuvion 51 mukaisesti sekä annettiin hallinnointia helpottava nimi, joka tässä tapauksessa oli laiterekisterissä määritetty laitetunnus. Tunnussanan piti olla tismalleen sama kuin aikaisemmin kytkimelle määritetty, muutoin NPS ei ota huomioon kytkimien pyyntöjä työaseman todentamiseksi. Tämän toimenpiteen jälkeen autentikontipalvelimella jäi enää tehtäväksi määrittää tarvittavat politiikat.



Kuvio 51. Kytkimelle annettu tunnussana

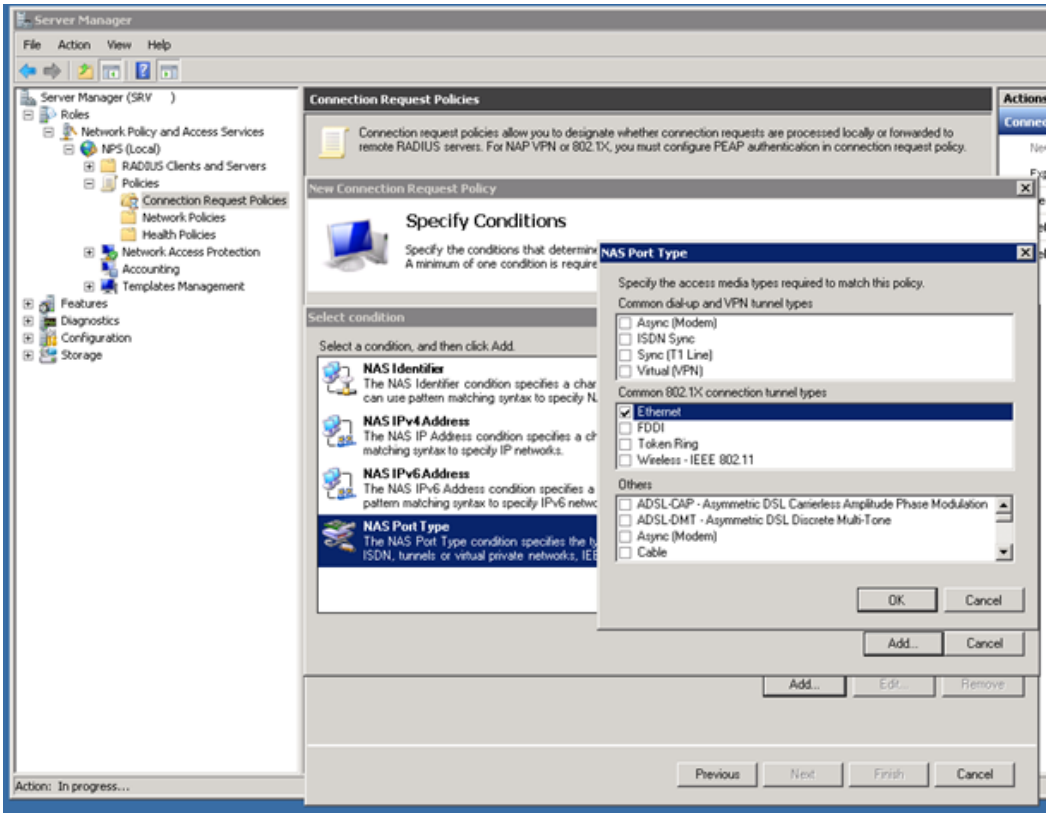
Connection Request Policies -hakemiston alle luotiin kuvion 52 mukaisesti uusi politiikka, mikä määrittää yhteyden muodostuksessa käytettävän yhteystyypin. Kyseiset politiikat käydään läpi määritetyssä järjestyksessä.



Kuvio 52. NPS Connection Request Policies

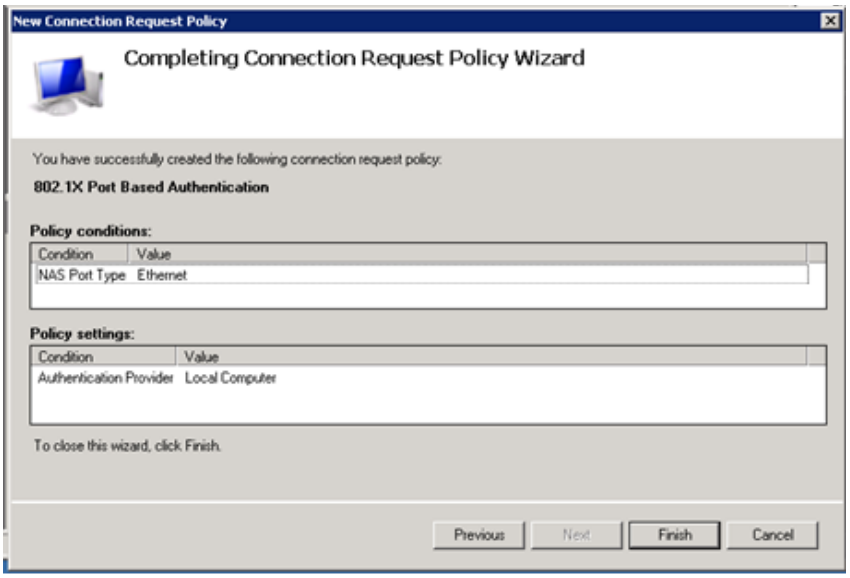
Mikäli vaaditut määritykset täyttyvät yhteyttä muodostaessa, hyväksyy NPS kytkimeltä saapuvan palvelupyynnön työaseman todentamiseksi. Tässä tapauksessa politiikka

määritettiin hyväksymään kuvion 53 mukaisesti kaikki päätelaitteilta tulevat pyynnöt, jotka ovat lähetetty Ethernet-verkosta.



Kuvio 53. Yhteystyypin vaatimukset

Alla olevassa kuviossa 54 on vielä yhteenveto politiikan asetuksista.

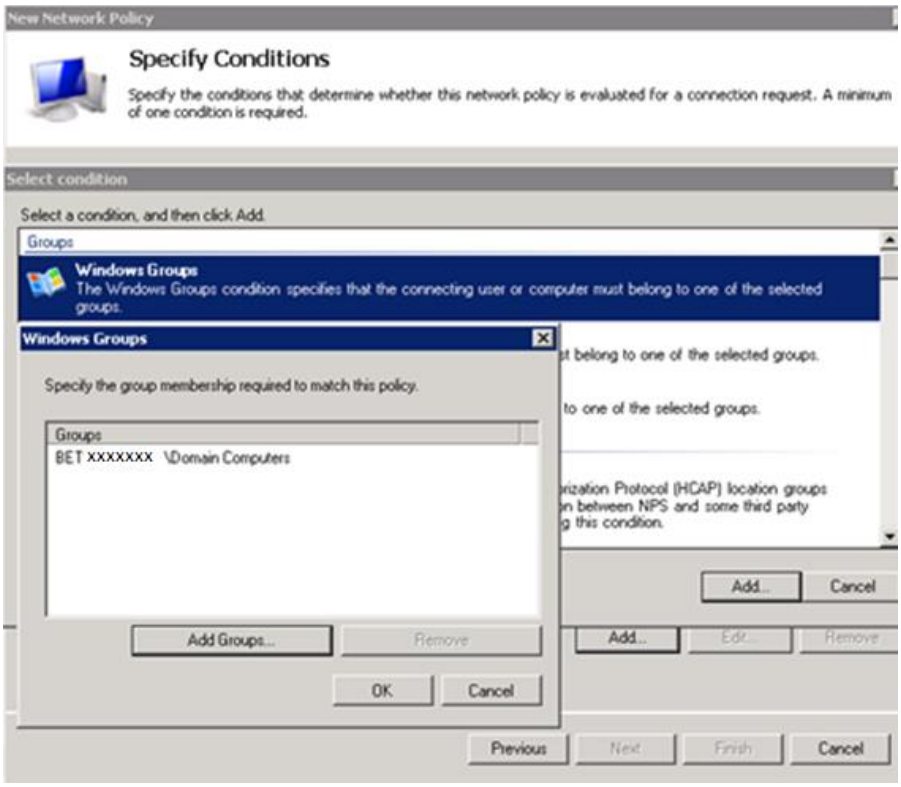


Kuvio 54. Yhteenveto Connection Request Policies -asetuksista

Network Policy määrittää ne vaatimukset, mitkä työaseman tulee täyttää ja mitä todennusmenetelmää käytetään. Lisäksi voidaan määrittää millaiset asetukset työasemalle annetaan onnistuneen todennuksen jälkeen. Kyseiset vaatimukset määritettiin Network Policies -hakemiston alle.

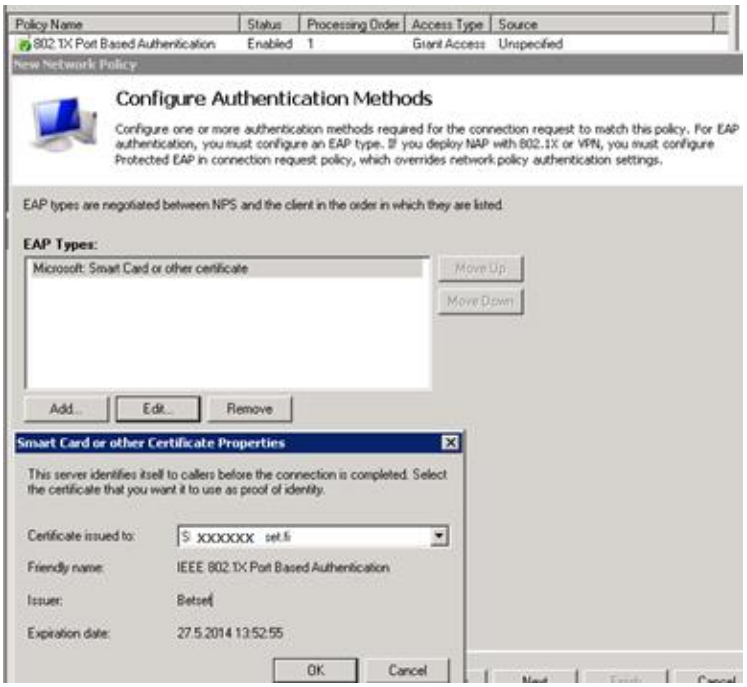
Yrityksen tietoverkkoon pyrkivien työasemien tulee olla osallisena toimialueessa ja niinpä vaatimukseksi asetettiin työaseman kuuluminen toimialueen ryhmään Domain Computers, kuvion 55 mukaisesti. Tämä ryhmä valittiin syystä, että kaikki toimialueeseen liittyvät työasemat kuuluvat automaattisesti tähän ryhmään. (Local and Domain Default Groups 2009.)

Kyseisellä menetelmällä voidaan luoda hyvinkin tarkkoja vaatimuksia mm. käyttäjistä sekä työasemista, mutta tässä työssä keskityttiin vain määrittämään työasemien vaatimukset. Syynä tähän oli estää tuntemattomien laitteiden sekä työasemien pääsy yrityksen tietoverkkoon.



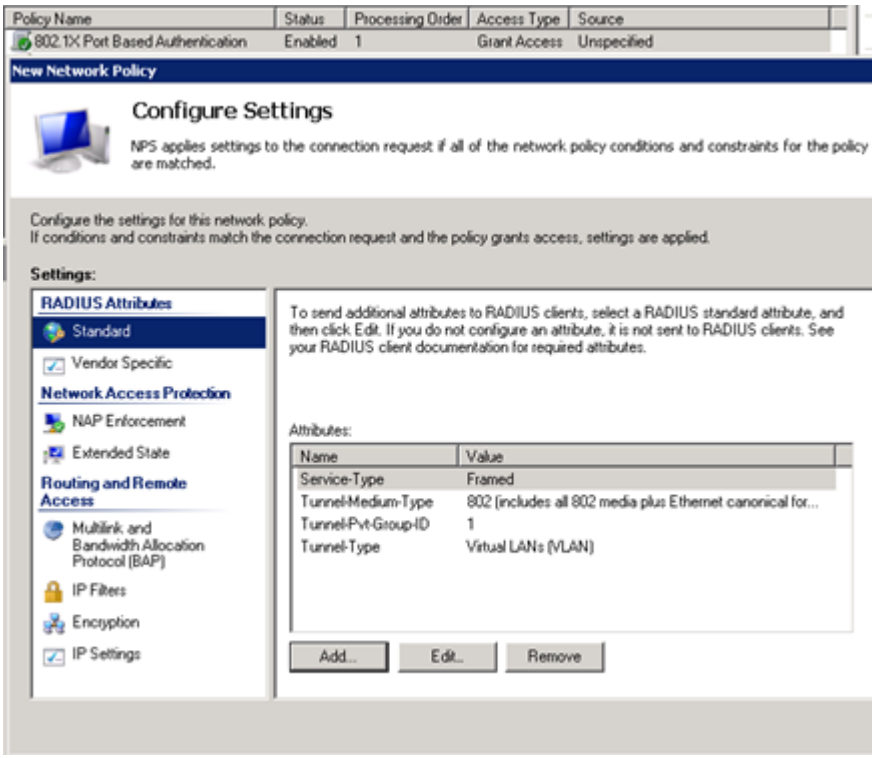
Kuvio 55. Työasemien vaatimukset

Koska yrityksessä oli jo valmiiksi toimiva julkisen avaimen infrastruktuuri, päätettiin käyttää työasemien todennuksessa varmenteita, kuvion 56 mukaisesti. Kyseinen valinta asettaa käytettäväksi EAP-metodiksi EAP-TLS:n. (Davies & Northrup 2008, 389.)



Kuvio 56. Käytettävä todennusmenetelmä ja palvelimen varmenne

Ylimääräiset ehdot tietoverkkoon pääsemiseksi ohitettiin. Seuraavaksi siirryttiin määrittelemään kuviossa 57 asetukset, jotka autentikointipalvelin lähettää kytkimelle mikäli työasemalta vaaditut ehdot täyttyvät. Kyseiset asetukset ovat RADIUS-attribuutteja ja niiden avulla voidaan muun muassa määrittää työasema tiettyyn VLANiin tai antaa yksittäiselle käyttäjälle kirjautumisoikeuksia verkkolaitteisiin. (Davies & Northrup 2008, 390-391.)



Kuvio 57. Kytimelle lähetettävät asetukset

Kuviossa 58 on Wireshark kuvankaappaus palvelimen sekä kytkimen väliltä, jossa palvelin hyväksyy työaseman pyynnön liittyä tietoverkkoon. Kyseinen viesti on RADIUS Access-Accept ja se sisältää aikaisemmin määritetyt attribuutit.

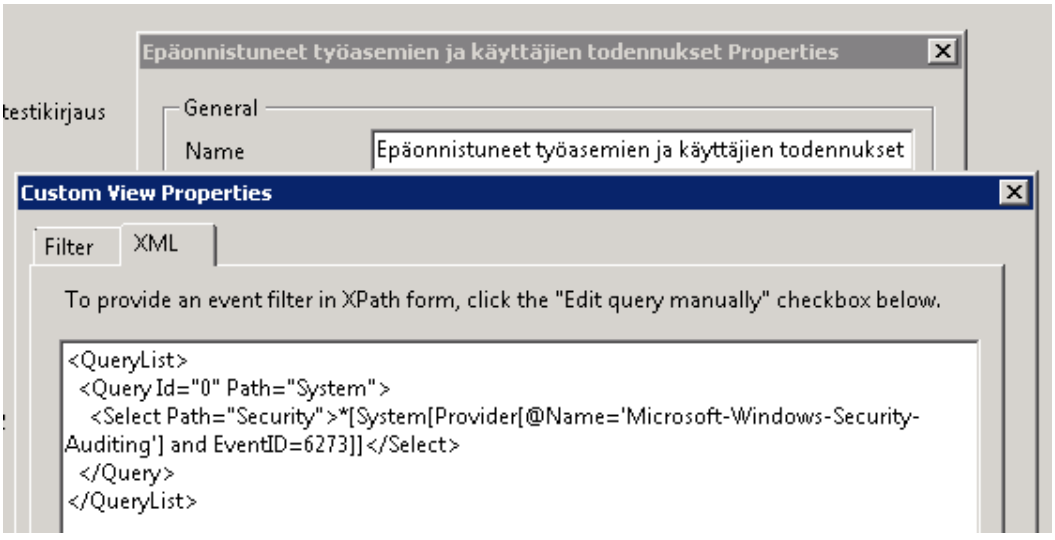
Kuvio 58. Työasemalle määritetyt asetukset

Tässä vaiheessa porttikohtainen todennus oli työasemia lukuun ottamatta valmis. Tapahtumienvalvontaan päätettiin vielä tehdä uusi kustomoitu näkymä, mikä näyttää kaikki epäonnistuneet todennusyritykset. Alla olevassa kuviossa 59 on havainnol-

listettu tämän kustomoidun näkymän toiminta. Kyseisestä näkymästä voidaan helposti selvittää mm. tietoverkkoon kuulumattomien laitteiden kirjautumisyritykset, kirjautumisyritykset aktiivilaitteisiin ja lisäksi näkymästä saataisiin tärkeätä tietoa työasemien testausvaiheessa.

Kuvio 59. Tapahtumienvalvonnan kustomoitu näkymä

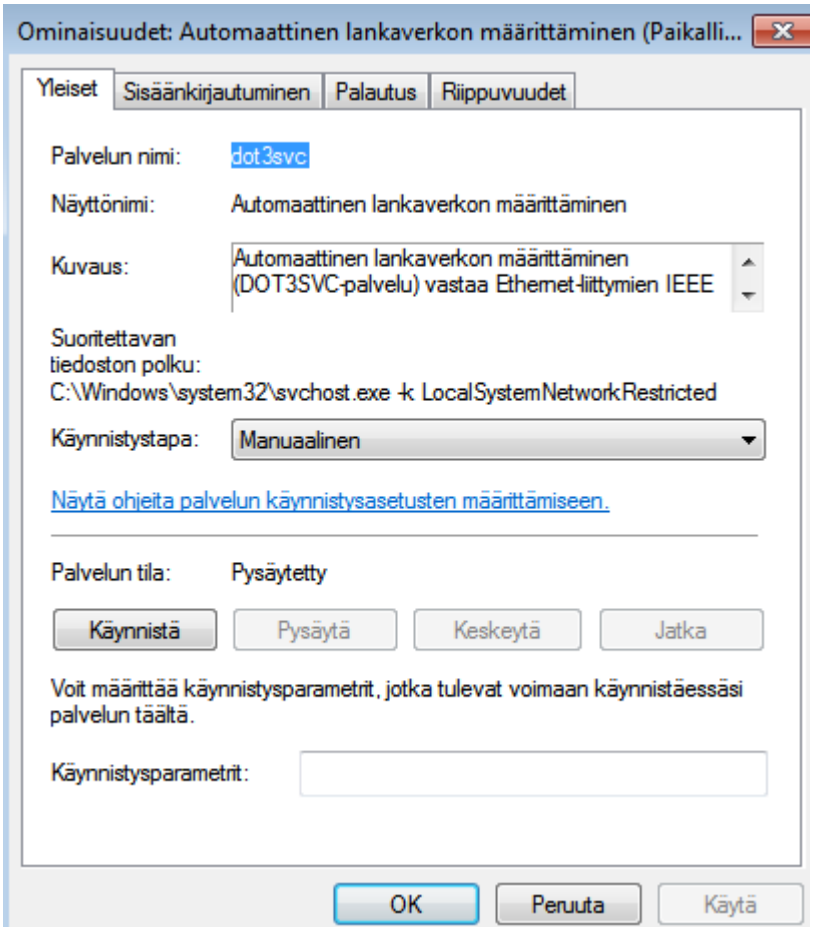
Kustomoitu näkymä on käytännössä eräänlainen suodatin, joka hakee määritetyin perustein tietyt tapahtumat lähdelokista ja kokoaa ne yhteen näkymään. Kuvion 60 mukainen suodatin määritettiin hakemaan Windowsin Security -lokista kaikki tapahtumat, jotka ovat tapahtumatunnisteella 6273. Kyseinen tunniste kuuluu Network Policy Serverille ja juuri tämä kyseinen tunniste kertoo epäonnistuneesta todennuksesta. (Event ID 6273 - NPS Authentication Status. 2008)



Kuvio 60. Luodaan uusi kustomoitu näkymä

6.4 Työasemien asetukset

Yrityksen työasemat oli varustettu Windows 7 Professional -käyttöjärjestelmällä ja niistä löytyi oletuksena tuki IEEE 802.1X:lle. Tämä ominaisuus on käytettävissä, kunhan työasemalla on käynnissä kuvion 61 mukainen dot3svc-palvelu lankaverkon määrittämiseksi. (802.1X-todennuksen ottaminen käyttöön 2014.)

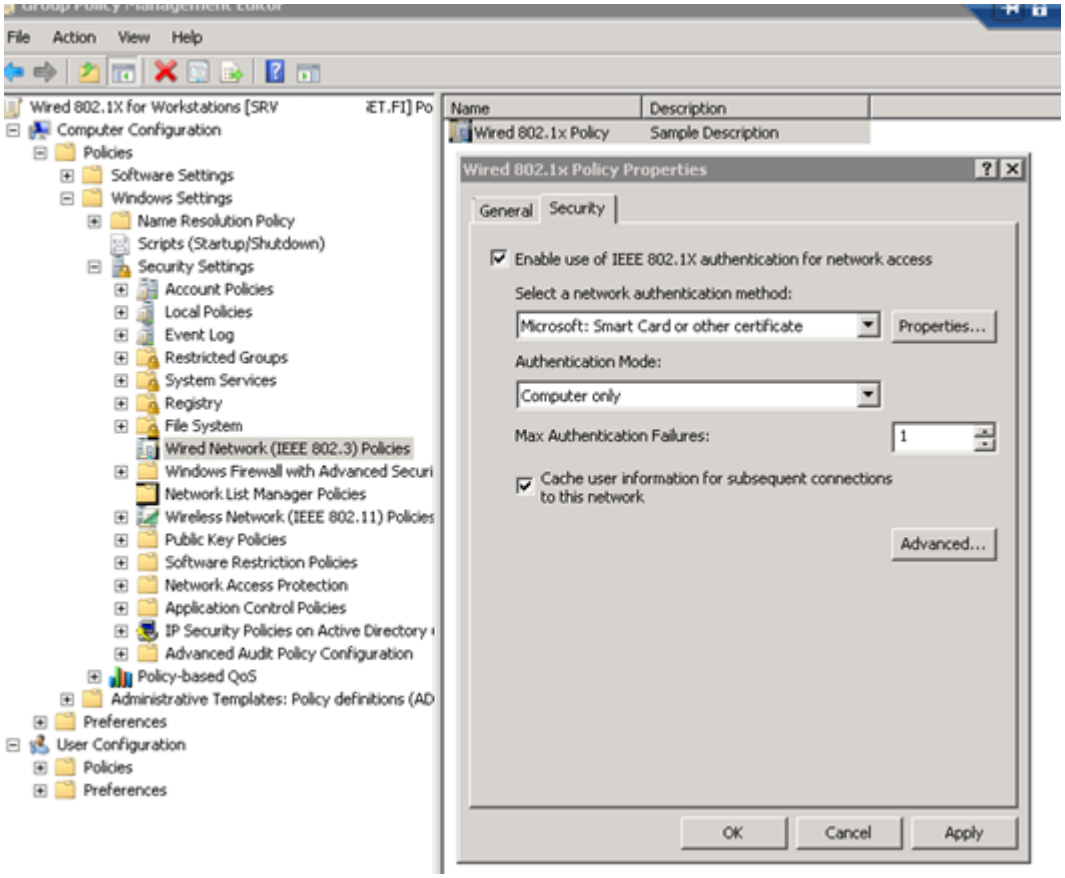


Kuvio 61. Dot3svc-palvelu

Ryhmäkäytäntöjen hallintakonsolilla (GPMC) luotiin kuviossa 62 uusi ryhmäkäytäntöobjektin (GPO). Tämä siitä syystä, että ensinnäkin ryhmäkäytännöt helpottavat huomattavasti työmäärää ja tarvittaessa asetuksia olisi helppo muuttaa. Tuolloin henkilöstön ei tarvitse käydä erikseen jokaisella työasemalla tekemässä tarvittavia muutoksia. Uudelle ryhmäkäytäntöobjektille annettiin nimeksi *Wired 802.1x for Workstations* ja sen alle määritettiin kaikki porttikohtaisessa todennuksessa tarvittavat asetukset.

Kuviossa 62 lisättiin Wired Network Policies -hakemiston alle uusi politiikka, joka määrittää työasemien verkkoasetukset. Avautuneesta valikosta otettiin käyttöön IEEE 802.1X-todennus ja koska autentikointipalvelimelle oli määritetty EAP-TLS, piti myös tästä valikosta ottaa käyttöön varmennepohjainen autentikointimenetelmä.

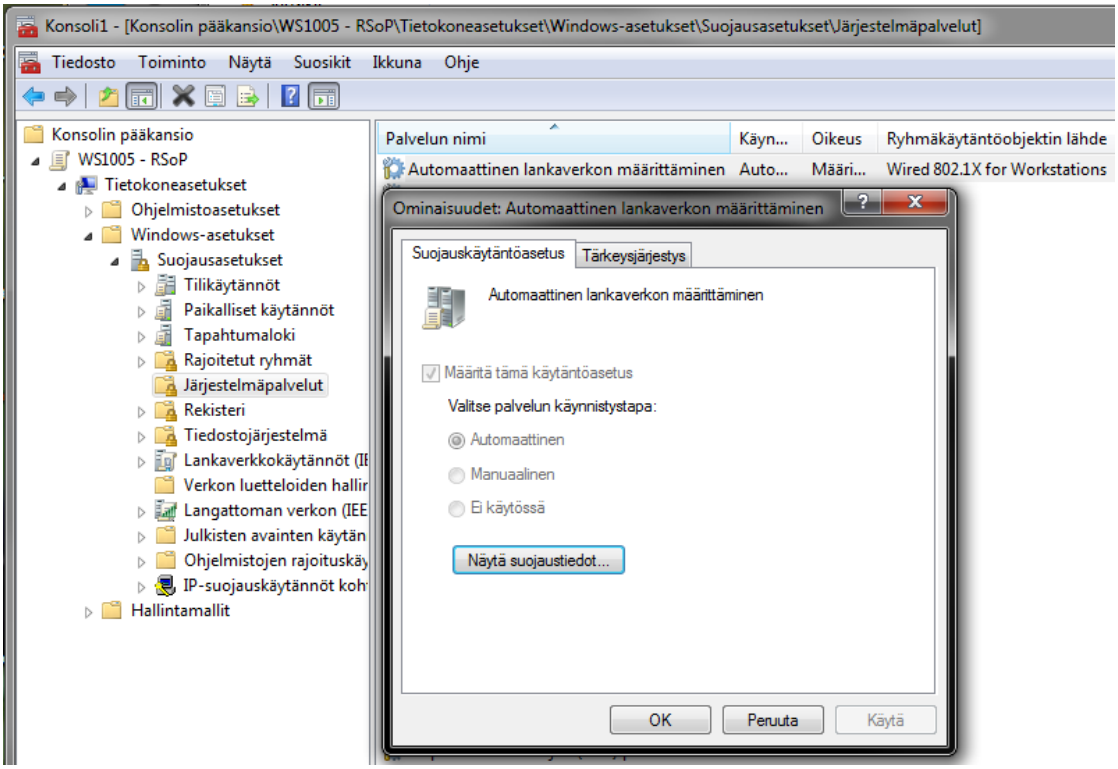
Varmenne valittiin properties-painikkeen takaa avautuneesta listasta ja lisäksi käyttäjille esitettävät ilmoitukset otettiin pois päältä. Käyttäjätietojen tallentaminen välimuistiin pidettiin vielä käytössä. (Configure 802.1X Wired Access Clients for EAP-TLS Authentication 2012. & Davies & Northrup 2008, 377-379.)



Kuvio 62. Ryhmäkäytäntöobjekti asetuksia varten

Uuden ryhmäkäytäntöobjektin toimivuus tarkistettiin käyttämällä kuviossa 63 RSoP-käytäntösuodatinta, millä voidaan tarkastella ryhmäkäytäntöobjektien määrittämiä asetuksia. Kuviosta voidaan todeta ryhmäkäytäntöobjektin toimivan määritettyihin työasemiin, sillä dot3svc-palvelu on nyt oletuksena päällä. Lisäksi ryhmäkäytäntöob-

jektin lähteenä on aikaisemmin luotu *Wired 802.1x for Workstations*. (Resultant Set of Policy.)

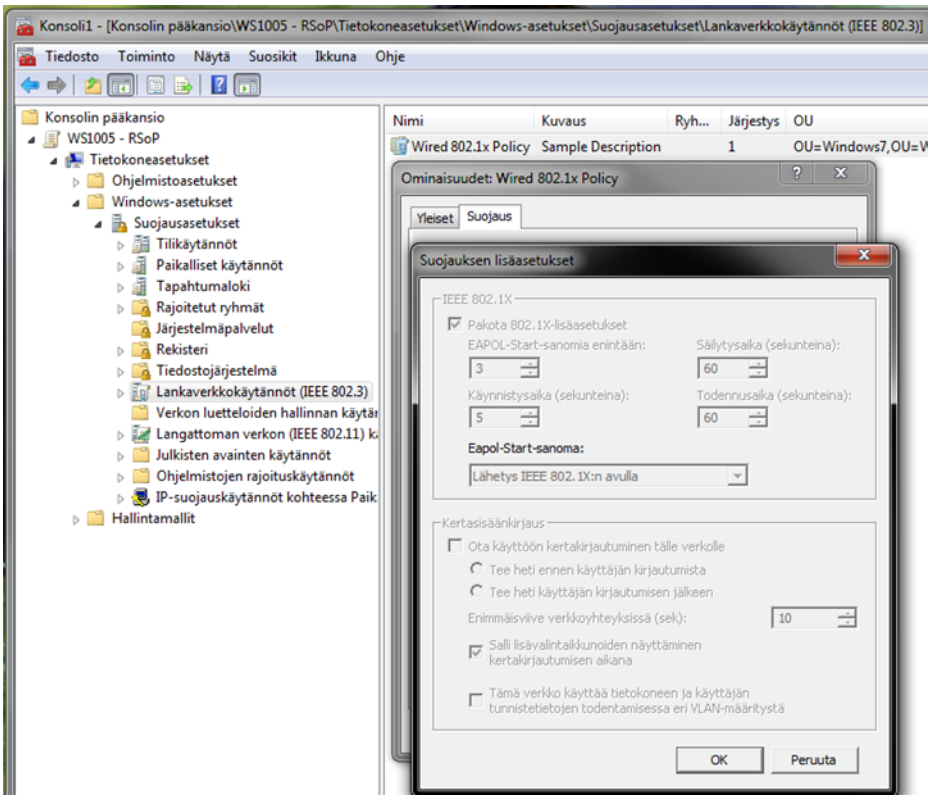


Kuvio 63. RSoP-käytäntösuodin

Ensisijaista autentikointipalvelinta käytettäessä, porttikohtainen todennus toimi loistavasti. Tehtävänannon mukaan, porttikohtaisen todennuksen piti toimia myös kaikissa mahdollisissa tilanteissa. Tämmöisiä tilanteita ovat esimerkiksi ensisijaisen autentikointipalvelimen vikaantuminen. Vikasietoisuutta testatessa, otettiin ensisijainen autentikointipalvelin pois päältä. Tämä tapahtui yksinkertaisesti pysäyttämällä NPS-palvelu hallintapaneelin kautta.

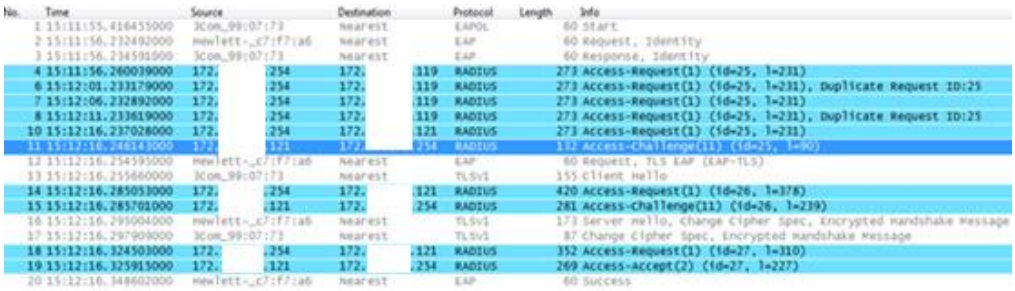
Käytettäessä toissijaista autentikointpalvelinta, ei työaseman todennus enää onnistunutkaan. Tilanteen selvittämiseksi otettiin käyttöön vielä toinen työasema ja todennusprosessin aiheuttama verkkoliikenne peilattiin kytkimeltä toiselle työasemalle. Ongelman syy löytyi lopulta Windows 7-käyttöjärjestelmän määrittämästä 802.1X:n todennusajasta, joka oli vain kahdeksantoista sekuntia. Tästä syystä työasema ei enää vastaa toissijaiselta autentikointipalvelimelta saamaansa EAP-TLS Request -viestiin.

Ongelman ratkaisemiseksi asetettiin todennusajan arvoksi kolmekymmentä sekuntia, tämä ei kuitenkaan vielä korjannut kyseistä ongelmaa, joten todennusaika nostettiin lopulta yhteen minuuttiin. Uudemmassa käyttöjärjestelmässä oli korotettu myös todennuksen säilytysaika yhteen minuuttiin, joten tämä muutettiin samalla vastaavaksi. Nämä edellä mainitut muutokset ovat todennettuna käytetyltä työasemalta ja näkyvät kuviossa 64. (Davies & Northrup 2008, 379.)



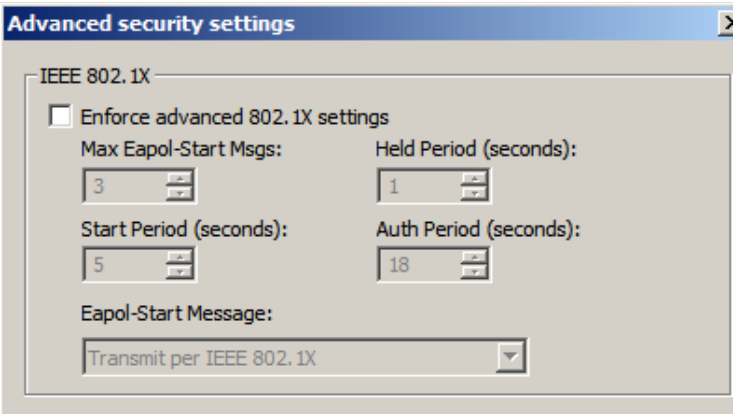
Kuvio 64. IEEE 802.1X-suojauksen lisäasetukset

Näiden muutosten jälkeen työaseman todentaminen toimi taas loistavasti, kuten myös kuvio 65 havainnollistaa. Ongelma olisi korjaantunut myös säätämällä kytkimien asetuksia, esimerkiksi kuinka kauan odotetaan vastausta ensisijaiselta autentikointipalvelimelta. Tässä vaiheessa kytkimien RADIUS-asetukset päätettiin pitää mahdollisimman alkuperäisinä.



Kuvio 65. EAP-TLS toimii

Virallista lähdettä Windows 7 -käyttöjärjestelmän suojauksen lisäasetuksille ei löytynyt, mutta useasta lähteestä löytyi vanhemmalle sekä uudemmalle käyttöjärjestelmälle käytetyt arvot. Lisäksi Windows 2008 R2:n esittämät arvot kuviossa 66 ovat tismalleen samat kuin vanhemmassa versiossa ja siinä ne vahvistetaan samalla menetelmällä oletusasetuksiksi. (Davies & Northrup 2008, 379; Advanced Security Settings for Wired and Wireless Network Policies 2012.)



Kuvio 66. Windows 2008 R2:n käyttämät oletusasetukset

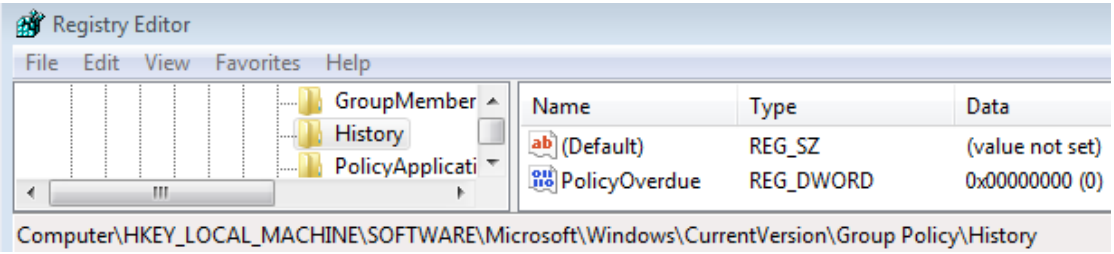
Kyseiset asetukset sekä niille testaamisessa löydetyt arvot ovat taulukossa 4 ja tarkemmat kuvaukset niiden käyttötarkoituksista ovat liitteessä 1.

Taulukko 4. 802.1X-suojauksen lisäasetukset

	Oletus		
	Windows 7	Windows 8	Betset
Max Eapol-Start Msgs	3	3	3
Held Period (seconds)	1	60	60
Start Perioid (seconds)	5	5	5
Auth Perioid (seconds)	18	30	60

Toissijaisen autentikointipalvelimen käyttö aiheutti vielä yhden ongelman, joka ilmeni vasta ryhmäkäytäntöjä testatessa. Porttikohtaisessa todennuksessa olevat työasemat eivät enää ottaneet vastaan ryhmäkäytännöillä määritettyjä asetuksia. Syynä tähän oli ryhmäkäytäntöjen prosessointi käyttöjärjestelmän käynnistysvaiheessa.

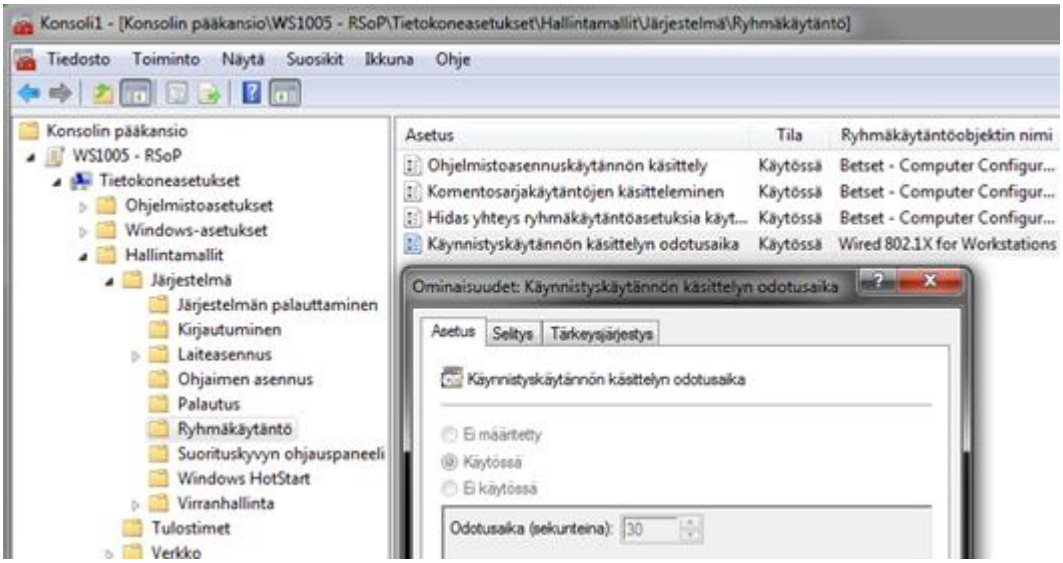
Mikäli käynnistysvaiheessa ei havaita toimivaa verkkoyhteyttä, epäonnistuu toimialueen domain controllerin (DC) löytyminen ja samalla myös ryhmäkäytäntöjen lataaminen epäonnistuu. Oletuksena Windows 7 -käyttöjärjestelmä laskee oman algoritminsa avulla keskimääräisen odotusajan ja tallentaa sen kuviossa 67 näkyvään Windowsin rekisteripolkuun. Kyseinen arvo riippuu useasta tekijästä ja vaihtelee työasemittain. Tästä syystä ongelman löytymiseen olisi voinut mennä hyvinkin kauan aikaa. (Windows 7 Clients intermittently fail to apply group policy at startup 2013.)



Kuvio 67. Ryhmäkäytäntöjen odotusaika

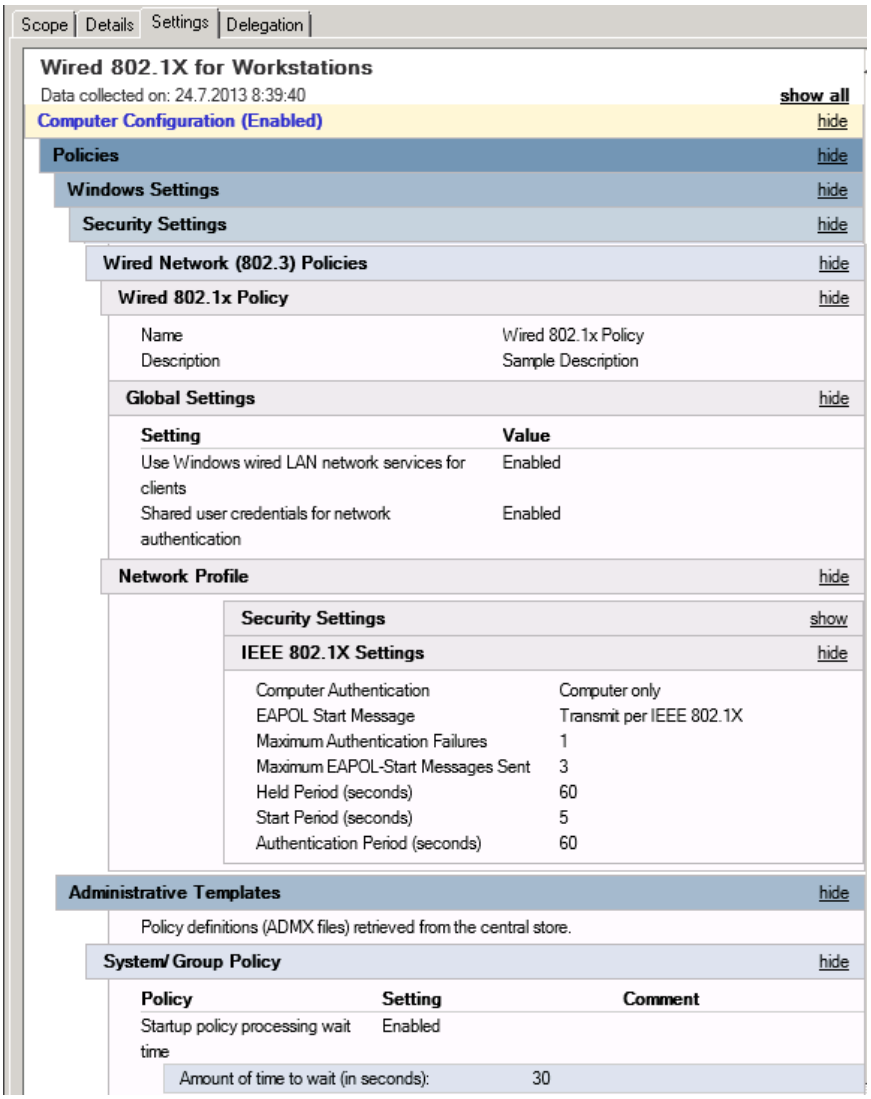
Kyseinen ongelma ratkaistiin lisäämällä ryhmäkäytäntöobjektiin asetus *Startup policy processing wait time* ja sen arvoksi asetettiin turvallinen kolmekymmentä sekuntia. Kyseinen toimenpide on todennettu alla olevassa kuviossa 68. Tämä arvo määrittää kuinka kauan käyttöjärjestelmä odottaa verkkoyhteyden muodostumista, kunnes käynnistysprosessia jatketaan. Mikäli verkkoyhteyttä ei saada muodostettua, lada-

taan ryhmäkäytäntöjen asetukset työaseman muistista. (Windows 7 Clients intermittently fail to apply group policy at startup. 2013.)



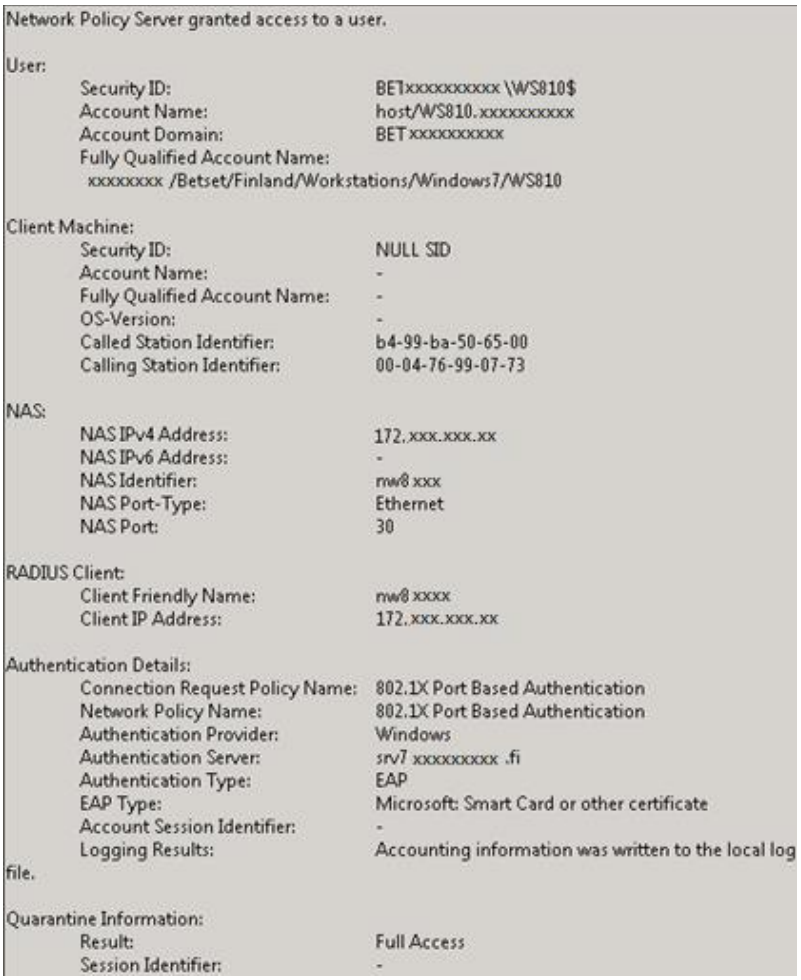
Kuvio 68. Käynnistyskäytännön käsittelyn odotusaika

Alla olevassa kuviossa 69 on vielä koostettuna yhteenveto käytetyn ryhmäkäytäntöobjektin asetuksista.



Kuvio 69. Käytetyn ryhmäkäytäntöobjektin asetukset

Lisäksi kuviossa 70 on todennettuna autentikointipalvelimen toiminta, sen käyttämät politiikat sekä työaseman onnistunut todennus. Kyseinen kuvio on kuvankaappaus autentikointipalvelimen tapahtumalokista.



Network Policy Server granted access to a user.

User:
- Security ID: BETxxxxxxxxx \WS810$
- Account Name: host/WS810.xxxxxxxxx
- Account Domain: BETxxxxxxxxx
- Fully Qualified Account Name: xxxxxxxx /Betset/Finland/Workstations/Windows7/WS810

Client Machine:
- Security ID: NULL SID
- Account Name: -
- Fully Qualified Account Name: -
- OS-Version: -
- Called Station Identifier: b4-99-ba-50-65-00
- Calling Station Identifier: 00-04-76-99-07-73

NAS:
- NAS IPv4 Address: 172.xxx.xxx.xx
- NAS IPv6 Address: -
- NAS Identifier: nw8 xxx
- NAS Port-Type: Ethernet
- NAS Port: 30

RADIUS Client:
- Client Friendly Name: nw8 xxxx
- Client IP Address: 172.xxx.xxx.xx

Authentication Details:
- Connection Request Policy Name: 802.1X Port Based Authentication
- Network Policy Name: 802.1X Port Based Authentication
- Authentication Provider: Windows
- Authentication Server: srv7 xxxxxxxxx .fi
- Authentication Type: EAP
- EAP Type: Microsoft: Smart Card or other certificate
- Account Session Identifier: -
- Logging Results: Accounting information was written to the local log file.

Quarantine Information:
- Result: Full Access
- Session Identifier: -

Kuvio 70. Työaseman onnistunut todennus

7 Yhteenveto

7.1 Työn tulokset ja niiden arviointi

Työ alkoi tutustumalla käytössä oleviin verkon aktiivilaitteisiin sekä niiden ominaisuuksiin. Tämä oli loistava aloitus työlle, koska en ollut aikaisemmin työskennellyt kyseisen laitevalmistajan tuotteilla. Tutustumisen jälkeen oli vuorossa tarvittavien komentojen metsästys porttikohtaisen todennuksen aktivoimiseksi. Kyseinen kohta oli hyvin opettavainen sekä antoi samalla uuden näkökulman laitevalmistajan tuotteisiin ja niiden mahdollisuuksiin.

Seuraavana vuorossa oli tutustuminen autentikointipalvelimeen. Kyseinen kohta vei suhteellisen paljon aikaa, koska Windowsin loogisuus poikkeaa huomattavasti koulussa opetettujen avoimien ohjelmistojen rakenteesta ja toimintatavasta. Microsoftin julkaisemien artikkeleiden ansiosta palvelimen asennus, toiminta, ja mahdollisuudet selvisivät suhteellisen nopeasti. Suurimmat ongelmat tässä vaiheessa olivat tarvittavien politiikkojen toimintojen ymmärtäminen ja kuinka saada verkkolaitteet keskustelemaan autentikointipalvelimen kanssa. Yksinkertaisin ja nopein vaihe tässä työssä oli työaseman valmistelu.

Tässä vaiheessa valmiina oli jo toimiva järjestelmä, joten mahdollisten vikatilanteiden sekä työympäristön tuomien ongelmien testaaminen saattoi alkaa. Tämä vaihe oli mielestäni kaikesta opettavaisin. Samalla se antoi myös käytännön kokemusta siitä, kuinka tämmöiset muutokset tulisi toteuttaa tuotantoympäristössä.

Mielestäni työ onnistui melko hyvin ja määritetyt vaatimukset saatiin toteutettua. Aluksi suunnitelmissa oli käsitellä myös muita porttikohtaisen todennuksen ohessa tehtyjä töitä, mutta siitä olisi tullut suhteellisen laaja kokonaisuus. Tästä johtuen päädyin käsittelemään tässä työssä vain porttikohtaista todennusta ja siihen liittyviä tekniikoita sekä osa-alueita.

Työn suorittaminen on mieluista, koska pääsin tavallaan jatkamaan koulussa suoritettua kurssia ja aihe oli ennestään tuttu. Muutenkin sain suhteellisen vapaat kädet

testailla ja rakentaa kyseistä järjestelmää. Esimiehen kanssa kommunikointi oli vaivatonta ja tarvittaessa sain aina opastusta sekä hyvin perusteltuja vastauksia.

7.2 Kehittämiskohteet

Vaikka porttikohtainen todennus onkin tehokas tapa estää tietoverkkoon kuulumattomien laitteiden käyttö, jäi kyseiseen ympäristöön vielä parannettavaa. Nykyinen järjestelmä toimii yksinkertaistettuna siten, että tietoverkkoon pyrkivän työaseman tulee kuulua toimialueen ryhmään.

Valvontakamerat, verkkotulostimet sekä muutama tuotannossa oleva vanhempi työasema ovat edelleen ilman todennusta. Tilanteen korjaamiseksi pitäisi tehdä lisätutkimusta, kuinka toteuttaa kyseiset korjaukset ja saada ne toimimaan nykyisessä järjestelmässä. Eräs ratkaisu olisi toteuttaa todennus MAC-osoitteiden perusteella.

Lähteet

802.1X-todennuksen ottaminen käyttöön. 2014. Microsoftin julkaisema artikkeli. Viitattu 26.4.2014. http://windows.microsoft.com/fi-fi/windows/enable-802-1x-authentication#1TC=windows-7

Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J & Levkowetz H. 2004. RFC3748. Viitattu 26.3.2014. http://tools.ietf.org/html/rfc3748

Aboba, B., Congdon, P., Roese, J., Smith, A., & Zorn, G. 2003. IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines. Viitattu 27.3.2014. http://tools.ietf.org/html/rfc3580

Adoba, D., Hurst, B. & Simon, D. 2008. The EAP-TLS Authentication Protocol. Viitattu 23.3.2014. http://tools.ietf.org/html/rfc5216

Advanced Security Settings for Wired and Wireless Network Policies. 2012. Microsoftin julkaisema artikkeli. Viitattu 27.4.2014. http://technet.microsoft.com/en-us/library/hh994696.aspx?ppud=4

ANSI/IEEE Std 802.2 Part 2: Logical Link Control. 1998. IEEE-organisaation julkaisema standardi. Viitattu 23.3.2014. http://standards.ieee.org/getieee802/download/802.2-1998.pdf

Blunk, L & Vollbrecht, J. 1998. PPP Extensible Authentication Protocol (EAP). Viitattu 22.3.2014. http://tools.ietf.org/html/rfc2284

Boeyen, S., Cooper, D., Farrell, S., Housley, R., Polk, W. & Santesson, S. 2008. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Viitattu 24.4.2014. http://tools.ietf.org/html/rfc5280

Choudhury, S. 2002. Public Key Infrastructure Implementation and Design. New York: M&T Books.

Command Line Interface Reference Guide. 2007. Hewlett-Packardin julkaisema käyttäjäopas. Viitattu 23.3.2014. http://ftp.hp.com/pub/networking/software/6200-5400-3500-CLI-k1201-Feb2007.pdf

Configure 802.1X Wired Access Clients for EAP-TLS Authentication. 2012. Microsoftin julkaisema artikkeli. Viitattu 27.4.2014. http://msdn.microsoft.com/en-us/library/dd759237.aspx

Davies, J. & Northrup T. 2008. Windows Server 2008 Networking and Network Access Protection (NAP). 1. p. Washington: Microsoft Press.

Ethernet Jumbo Frames. 2009. Ethernet Alliancen julkaisema artikkeli. Viitattu 7.4.2014. http://www.ethernetalliance.org/wp-content/uploads/2011/10/EA-Ethernet-Jumbo-Frames-v0-1.pdf

Event ID 6273 - NPS Authentication Status. 2008. Microsoftin julkaisema artikkeli. Viitattu 25.3.2014. http://technet.microsoft.com/en-us/library/cc735399%28v=ws.10%29.aspx

Funk, P. & Blake-Wilson, S. 2008. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). Viitattu 24.3.2014. http://tools.ietf.org/html/rfc5281

Garman, J. 2003. Kerberos The Definitive Guide. USA: O'Reilly.

Hakala, M. & Vainio, M. 2005. Tietoverkon rakentaminen. Jyväskylä: Docendo Finland Oy.

Heikkilä, S. 2013. Controller. Betset Oy. Harjoittelun yhteydessä suoritettu haastattelu.

Historia. 2014. Yrityksen julkaisemaa tietoa konsernin historiasta. Viitattu 14.5.2014. http://www.betset.fi/fi/info/historia

Hämeen-Anttila, T. 2003. Tietoliikenteen perusteet. Jyväskylä: Docendo Finland Oy.

IEEE Std 802.3-2012. 2012. IEEE-organisaation julkaisema standardi. Viitattu 21.3.2014. http://standards.ieee.org/getieee802/download/802-2001.pdf

Jaakohuhta, H. 2005. Lähiverkot – Ethernet. 4. uud. p. Helsinki: Edita.

Josefsson, S., Palekar, A., Salowey, J., Simon, D., Zhou, Hao. & Zorn, G. 2004. Protected EAP Protocol (PEAP) Version 2. Viitattu 25.3.2014. http://tools.ietf.org/html/draft-josefsson-pppext-eap-tls-eap-10

Kamath, V. & Palekar, A. 2004. Microsoft EAP CHAP Extensions. Viitattu 22.3.2014. http://tools.ietf.org/id/draft-kamath-pppext-eap-mschapv2-01.txt

Kamath, V., Palekar, A. & Wodrich, M. 2002. Microsoft's PEAP version 0 (Implementation in Windows XP SP1). Viitattu 26.3.2014. http://tools.ietf.org/html/draft-kamath-pppext-peapv0-00

Konsernirakenne. 2014. Yrityksen julkaisemaa tietoa konsernin rakenteesta. Viitattu 14.5.2014. http://www.betset.fi/fi/info/konsernirakenne

Local and Domain Default Groups. 2009. Microsoftin julkaisema artikkeli. Viitattu 24.4.2014. http://technet.microsoft.com/en-us/library/dd728026%28WS.10%29.aspx

Minasi, M. 2010. Mastering Windows Server 2008 R2. Indiana: Wiley Publishing.

MS-CHAP v2. Microsoftin julkaisema artikkeli. Viitattu 23.3.2014. http://technet.microsoft.com/en-us/library/cc957983.aspx

Network Policy and Access Services. 2014. Microsoftin julkaisema artikkeli. Viitattu 24.4.2014. http://technet.microsoft.com/en-us/network/bb545879.aspx

PEAP Overview. 2012. Microsoftin julkaisema artikkeli. Viitattu 26.3.2014. http://technet.microsoft.com/library/cc754179.aspx

Port-Based Network Access Control. 2010. IEEE-organisaation julkaisema standardi. Viitattu 27.3.2014. http://standards.ieee.org/getieee802/download/802.1X-2010.pdf

Potter, D & Zamick, J. 2002. PPP EAP MS-CHAP-V2 Authentication Protocol. Viitattu 22.3.2014. http://tools.ietf.org/html/draft-dpotter-pppext-eap-mschap-01

Release Notes: Version F.05.72 Softwarefor the ProCurve Series 2300 and 2500 Switches. 2009. Hewlett-Packardin julkaisema käyttäjäopas. Viitattu 23.3.2014. http://cdn.procurve.com/training/Manuals/2300_2500-RelNotes-F0572-59903102.pdf

Resultant Set of Policy. Microsoftin julkaisema artikkeli. Viitattu 27.4.2014. http://technet.microsoft.com/en-us/library/cc772175.aspx

Rigney, C., Rubens Merit, A., Simpson Daydreamer, W. & Willens Livingston, S. 2000 RFC2865 Remote Authentication Dial In User Service (RADIUS). Viitattu 22.3.2014. http://tools.ietf.org/html/rfc2865

Rigney Livinstong, C. 2000. RFC2866 RADIUS Accounting. Viitattu 22.3.2014. http://tools.ietf.org/html/rfc2866

Rivest, R. 1992. The MD5 Message-Digest Algorithm. Viitattu 22.3.2014. http://tools.ietf.org/html/rfc1321

Simpson, W. 1996. PPP Challenge Handshake Authentication Protocol (CHAP). Viitattu 22.3.2014. http://tools.ietf.org/html/rfc1994

Smith, E. 2001. Authentication From Passwords to Public Keys. USA: Addisson-Wesley.

Tomes, T. & Baggett, M. 2011. Official Release: eapmd5crack.py. Viitattu 23.3.2014. http://lanmaster53.com/2011/04/official-release-eapmd5cracker-py/

Windows 7 Clients intermittently fail to apply group policy at startup. 2013. Microsoftin julkaisema artikkeli. Viitattu 27.4.2014. http://support.microsoft.com/kb/2421599

Liitteet

Liite 1. IEEE 802.1X Advanced Security Settings

Asetus	Kuvaus
Max Eapol-Start Msgs	Kuinka useasti lähetetään EAPOL-Start -viesti, mikäli alkuperäiseen ei saada vastausta.
Held Period (seconds)	Kuinka kauan odotetaan uudelleenyritystä epäonnistuneen todennuksen jälkeen.
Start Period (seconds)	Tämä asetus määrittää ajanjakson, kuinka kauan odotetaan mahdollista uudelleenyritystä.
Auth Period (seconds)	Määrittää kuinka kauan odotetaan ennen uudelleenyritystä, mikäli porttikohtainen todennus havaitaan.